UNIVERSITY OF CALIFORNIA
Los Angeles

Rare Event Prediction with Mortgage Lead Data

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Applied Statistics

by

Kaleb Julian Erickson

2019

ABSTRACT OF THE THESIS

Rare Event Prediction
with Mortgage Lead Data

by

Kaleb Julian Erickson
Master of Applied Statistics
University of California, Los Angeles, 2019
Professor Guido Montufar Cuartas, Chair

LeadPoint Inc. is a digital marketplace for refinance mortgage leads. These leads are purchased by lenders who then reach out to contact the lead in an attempt to refinance their mortgage. LeadPoint is interested in creating a predictive model that will identify leads that have a higher propensity to become a funded loan. This paper describes the process of using lead data from LeadPoint to create a model that predicts which leads are most likely to fund. The lead data is extremely imbalanced, with only 0.55% of the leads listed as a funded loan. The scarcity of funded loans in the data qualify a funded loan as a rare event. Since the vast majority of the leads do not end up funding, it is extremely difficult to accurately predict how any given lead will end up. This paper considers three different methods for dealing with this rare event data and compares them to a baseline logistic regression model. These additional methods include a method called Rare Event Logistic Regression, Gradient Boosted Decision Trees (specifically using a technology called CatBoost), and a data augmentation method called Synthetic Minority Oversampling Technique (SMOTE). The results showed that the rare event logistic regression model trained on the original data had the best performance, although the results were only slightly better than the logistic regression model. This rare event logistic regression model is able to identify a subset of leads with a fund rate of 0.90%. While this new fund rate is only 67% better than the original dataset, this is a very good model and represents expanded business opportunities for LeadPoint as well as a potential 19% immediate increase in company revenue.

The thesis of Kaleb Julian Erickson is approved.

Frederic R. Paik Schoenberg

Maryam M. Esfandiari

Guido Montufar Cuartas, Committee Chair

University of California, Los Angeles

2019

For my parents, who inspired my love of learning.

And my wife, who encourages me to keep learning.

TABLE OF CONTENTS

1 Introduction **1**

1.1 Background 1

1.2 Mortgage Leads 2

1.3 Rare Events 3

2 Modeling Methods **5**

2.1 Synthetic Minority Over-Sampling Technique (SMOTE) 5

2.2 Logistic Regression 9

2.3 Rare Event Logistic Regression 9

2.4 Gradient Boosted Decision Trees - CatBoost 11

3 Feature Exploration and Selection **16**

3.1 Exploratory Data Analysis 17

3.2 Feature Selection 35

3.2.1 Chi-Squared Test 36

3.2.2 Information Value 36

4 Methods for Model Evaluation **41**

4.1 Performance Metrics 41

4.2 Splitting the Data 43

4.3 Choosing the Prediction Threshold 44

4.4 SMOTE Dataset 45

5 Model Results **47**

5.1 Training Results 47

5.2 Final Model Equation and Coefficients 54
5.3 Final Model Evaluation 59

6 Modeling Outlook 62
6.1 Modeling Method Comparison 62
6.1.1 Further Uses for Rare Event Logistic Regression 64
6.2 Limitations 64
6.3 Multiple Classification Alternative 66

7 Conclusion 69

Appendix A 70

Appendix B 75

Appendix C 78

Appendix D 82

References 88

LIST OF FIGURES

3.1 Histogram showing the distribution of leads by hour. 17

3.2 Plot of the fund rate by hour. 18

3.3 Histogram of values for Loan Value in the lead dataset. 21

3.4 Histogram of values for LTV in the lead dataset. 24

3.5 Histogram of values for the Add Cash variable. 27

3.6 Histogram of values for the Add Cash variable. 29

3.7 Histogram of values for the Income variable. 34

3.8 Plot of the average fund rate by income. 35

5.1 Model Results on Feature Set 1. 48

5.2 Model Results on Feature Set 2. 49

5.3 Model Results on Feature Set 3. 49

5.4 Model Results on Feature Set 4. 50

5.5 Model Results on Feature Set 5. 50

5.6 Model Results on Feature Set 1. 52

5.7 Model Results on Feature Set 2. 52

5.8 Model Results on Feature Set 3. 53

5.9 Model Results on Feature Set 4. 53

5.10 Model Results on Feature Set 5. 54

6.1 Histogram of the prediction values from the Relogit Model on the test set with the original cutoff point. 67

6.2 Histogram of the Relogit Model prediction values with two cutoff points to classify leads based on their predicted fund rate. 68

D.1 An example of how the model summary looks with the feature coefficient estimates. 86

D.2 An example of what the output histogram might look like, with each column and row labeled. 86

D.3 A snippet of the lead data from LeadPoint. 87

LIST OF TABLES

1.1 The distribution of funded loans within the dataset of leads purchased from LendingPoint. 3

2.1 Example of SMOTE-N 8

3.1 Lead count and fund rates for leads purchased during each hour of the day . . . 19

3.2 Lead count and fund rate for the groupings of purchase hour. 20

3.3 Six number summary for the Loan Value variable 21

3.4 Lead count and fund rate for the proposed Loan Value groupings 22

3.5 A common example of Credit Grade guidance found on online mortgage forms. . 23

3.6 Lead count and fund rate for each Credit Grade. 23

3.7 Lead count and fund rate for LTV Ratio. 24

3.8 Lead count and fund rate for the newly proposed grouping of LTV Ratios. . . . 25

3.9 Six number summary for the Add Cash variable. 26

3.10 Lead count and fund rate of proposed grouping for Add Cash. 27

3.11 Lead count and fund rate of proposed grouping for Add Cash. 28

3.12 Six number summary for the 1st Mortgage Interest Rate variable. 28

3.13 Lead count and fund rate of proposed grouping for Interest Rate. 29

3.14 Lead count and fund rate of for the values of the 2nd Mortgage variable. 30

3.15 Lead count and fund rate of proposed grouping for Number of Mortgage Lates. . 31

3.16 Lead count and fund rate of proposed grouping for VA Status. 31

3.17 Lead count and fund rate of for the values of the 2nd Mortgage variable. 32

3.18 Lead count and fund rate of for the values of Loan Type. 33

3.19 Six number summary for the Income variable. 34

3.20 Lead count and fund rate of proposed grouping for Income. 35
3.21 Results of the Chi-Squared test on each of the 13 variables. 37
3.22 Demonstration of how to calculate the Information Value of the Credit Grade feature. 38
3.23 Information Value for each feature. 39
3.24 List of the five proposed feature sets. 40

4.1 Distribution of leads and fund rate between the three splits of the dataset. . . . 43

5.1 The coefficient estimates and standard errors for each feature in the final Relogit model. 56
5.2 The final metric evaluation results for the Relogit Model on the test set. 60
5.3 The confusion matrix from the final results of the Relogit Model on the test set. 60

6.1 Table showing the fund rate and Improvement Metric for each of the tiered classifications. 67

B.1 Table showing the difference in the distribution and fund rate between the Original and SMOTE datasets for each variable in Feature Set 5. 75

C.1 The results of each model trained on the original dataset followed by the SMOTE dataset with all categorical variables. 78
C.2 The results of each model trained on the original dataset followed by the SMOTE dataset using the four numeric variables instead of their categorical groupings. . 80

ACKNOWLEDGMENTS

I would like to thank all of the people that supported me and guided me through this process.

Thank you to Tudor for mentoring me as an analyst.

Thank you to Dhanas for all of the machine learning tips.

Thank you to Luke for keeping me sane with pizza and Halo.

Thank you to Tina for letting me crash at her place so often.

And thank you to Mackenzie for always being there for me.

CHAPTER 1

Introduction

1.1 Background

There is approximately $15.5 trillion worth of mortgage debt in the United States and this value is increasing each year [1]. As home values increase and interest rates fluctuate over time, there are a myriad of reasons for consumers to want to refinance their home mortgage. The most common reason for refinancing is to negotiate a lower interest rate on the loan, especially for consumers with adjustable rate mortgages. Many others refinance their loan in order to leverage the equity in their home to receive a cash payout that will be repaid over time.

Mortgage lenders have several options for finding consumers interested in refinancing their loan. The bulk of their traffic is organic – people who walk into a brick and mortar location to discuss their refinancing options. Many people prefer to work with their local bank or credit union for simplicity. However, when organic traffic slows, many lenders turn to online lead generation companies who sell "leads" on consumers who want to refinance their mortgage. A lead is simply a collection of information about a consumer who has expressed interest in purchasing a product or service; here, they have specifically expressed interest in refinancing their current mortgage. This means that they have clicked on some sort of marketing on the internet or in their email which led them to fill out an online form with information about their loan. This information is bundled as a lead, which is then sold to mortgage lenders who will try to contact the consumer to move forward with the refinancing process.

The downside of purchasing online leads is that most mortgage refinance leads are not

very likely to result in a customer following through and refinancing their loan. It follows that lenders would be willing to pay a higher price for leads that are shown to be more likely to fund than the average mortgage lead. These higher quality leads would require less time to work and would improve the efficiency of the lender's process. For the lead generation company, this would increase prices on lead inventory, encourage current lenders to continue buying leads, and work to garner attention from other lenders interested in buying more efficient leads.

LeadPoint Inc. is a lead generation company based in Los Angeles, CA. They have been operating since 2004 and sell mortgage leads to many of the top lenders in the United States. LeadPoint would like to improve their lead generation process by using a predictive model to identify leads that have a higher propensity to end up as a refinanced loan, also referred to as a funded loan. Any leads that are shown to be more likely to fund can be sold for a higher price and mortgage lenders would be willing to pay more for these higher quality leads. This could also help attract smaller lenders who require higher efficiency leads in order to make a profit. This paper describes the process of using lead data from LeadPoint in order to created a model that best accomplishes this goal.

1.2 Mortgage Leads

A mortgage lead consists of 13 different data points that contain information about the loan in question, as well as some information about the consumer's relevant financial history. All of this data is self-reported, so there is some chance of inaccurately reported information in each data point. These data points are a mix of numeric, categorical, and binary values, many of which will require consideration as to how they are engineered for use in training a model. The response variable is a binary flag that indicates whether that lead became a funded loan. This data only includes lead purchased from LeadPoint, so there are no leads in this dataset that did not have a chance to become a funded loan.

The data from LeadPoint Inc was collected over a 5 month period in 2018 and high-level

Table 1.1: The distribution of funded loans within the dataset of leads purchased from LeadPoint.

Total Leads	Total Funded	% Funded
494,025	2,714	0.55%

information about the data can be seen in Table 1.1. It is readily apparent that the fund rate on these leads is incredibly low, with only 1 lead funding out of every 182 leads. Due to the rarity of funded loans in the data, a lead that becomes a funded loan is considered to be a rare event.

1.3 Rare Events

Training a model on rare event data generally results in a model that is heavily biased toward the majority class. For example, a model could predict that none of the leads will fund and it will be 99.45% accurate, even though it didn't actually predict anything. By any other standard, that would be a high level of accuracy, but in this case it is a complete failure. Determining the correct method for model evaluation is an important part of the modeling process and is discussed further in section 4.1.

The other difficulty, especially with this lead data, is that there are thousands of leads with the same exact attributes, but only a handful of them actually became funded loans. In much of this data, there is no difference in features between the leads that ended up funding and the ones that did not. This makes accurate classification very difficult and models trained on this data will likely have a high rate of false-positives.

Rather than attempting to perfectly predict every lead, the goal of this model is to identify a subset of leads that are more likely to fund than leads from the original dataset. Even though the fund rate in that subset will still be very low, any fund rate higher than the 0.55% fund rate of the original dataset can directly result in additional revenue for LeadPoint.

With these challenges in mind, several different models were fit to the data; including a logistic regression model (see section 2.2), a rare event logistic regression model (see section 2.3), and a gradient boosted decision tree (done using the CatBoost methodology - see section 2.4). In addition to these three model types, a technique called Synthetic Minority Over-Sampling (or SMOTE) was used to create an artificially balanced dataset (see section 2.1). Each of the models were trained on this artificial dataset as well as the original, unaltered dataset to see which results in a better classifier. The logistic regression model is considered as a baseline to compare with the other methods to understand whether they actually improve the predictions on this rare event data. Unfortunately, the nature of this data makes it unlikely that any of these methods will completely outperform the others.

Ultimately, the rare event logistic regression model trained on the original dataset had the best performance, as it was able to identify a subset of leads that are 67% more likely to fund than any given lead in the original set of leads (see section 5.3 for the final model results).

CHAPTER 2

Modeling Methods

With such a small number of successfully funded leads in the data, this dataset is significantly imbalanced toward leads that don't fund. Strong imbalances in a training set can potentially lead to bias and overfitting in the resulting model. Fortunately, this is a large dataset with a total of 2,714 successful entries and 494,025 total entries, so that should help improve the performance of each model. Even so, the imbalanced nature of this dataset cannot be ignored.

In solving this problem, the first priority is to correctly identify as many leads from the positive class as possible. Fortunately, erroneously classifying negative leads as positive is only a moderately severe mistake. Further discussion of the model evaluation techniques can be found in section 4.

This chapter explains the three different methodologies that were utilized in this paper to mitigate the effects of the data imbalance and create the most effective model to predict which leads will fund.

2.1 Synthetic Minority Over-Sampling Technique (SMOTE)

The first methodology used to deal with the imbalance in the dataset is not a modeling technique, but rather a pre-processing method called Synthetic Minority Over-Sampling Technique (SMOTE). SMOTE was first proposed by Chawla et al. in a 2002 paper titled "Synthetic Minority Over-Sampling Technique (SMOTE)" [2]. In this paper they discuss the problems that arise with imbalanced data; particularly that the cost of misclassifying a positive example as false is often more costly than identifying a false example as positive, and

that classifiers trained on imbalanced data can obtain a misleading high predictive accuracy by simply always predicting the positive class. They review other methods for balancing data, specifically under and over-sampling techniques as well as the use of non-uniform weighting, then they propose their own method of balancing called SMOTE.

SMOTE utilizes both the under-sampling of the majority class with a special form of over-sampling the minority class to artificially balance the data and improve model predictions. SMOTE aims to increase the sensitivity of a classifier to the minority class, thus producing more accurate results. To measure the effectiveness of SMOTE, they experimented on nine different datasets of various size and levels of imbalance. Three different algorithms were applied to each of these datasets: C4.5, Ripper, and the Naive Bayes Classifier. These classifiers were trained with SMOTE as well as other techniques for dealing with imbalanced data (including plain under-sampling, varied loss ratios, and varied class priors). The results of the classifiers were compared using the Area Under the ROC curve (AUC) and it was found that the SMOTE approach had the highest accuracy in 44 out of 48 experiments performed.

The special form of over-sampling involves creating "synthetic" or artificial entries in the minority class that are used in training a learner. These synthetic entries do not exist in the original dataset, but are created from similar entries found using the K Nearest Neighbors algorithm. Creating synthetic entries allows for increasing the size of the minority class without having to re-sample replicated entries from the minority class of the original data. Even though the synthetic entries aren't necessarily "real", they are created with features similar to the actual minority class entries, thus helping to generalize the learnings of a model to make better predictions.

Instead of creating enough synthetic entries to match the size of the majority class, the majority class is also reduced in size through random under-sampling. Entries are randomly removed from the majority class until it becomes a specified size with respect to the size of the over-sampled minority class. Since the majority class is so heavily over-represented, removing some of the entries likely does not remove any important information from the data. By combining both the synthetic over-sampling of the minority class with the random

under-sampling of the majority class, the result is a dataset that is no longer biased toward one or the other. This artificial balance helps create less-biased learners when fitting a model.

Creating the synthetic entries for SMOTE involves use of K Nearest Neighbors to find similar entries in the minority class. This is done by first, finding the k nearest neighbors for each entry in the minority class. In calculating the nearest neighbors, first the feature values need to be normalized, then the distance between entries is measured using the sum of the absolute value of the differences between features. One of these k neighbors is chosen at random and compared to the original entry. The difference between their feature values is calculated, then multiplied by a random number between 0 and 1, and added back to the original entry. The result is a random point between the feature values of the two entries. This process is done for each feature and the resulting values make up a synthetic entry. (Note that the synthetic entries are not included when searching for the nearest neighbors - this process only uses the real data in order to maintain its integrity). By repeating this entire process for every entry in the original minority class, the result is a synthetic dataset of comparable size to the real minority class. If a bigger synthetic dataset is needed, the process can be repeated with multiple neighbors to create multiple synthetic entries.

The dataset in this problem primarily consists of categorical variables, which don't have geometric distances between each other. In the case of nominal variables, a modified approach is applied called SMOTE-N (Chawla et al., 2002). This only affects how the synthetic entries are created, the over-sampling and under-sampling processes are the same as described in SMOTE. Start with the first entry in the minority class, i. This entry is then compared with every other entry in the minority class to determine which feature values that i has in common with each other entry. The k entries that have the most values in common with i are determined to be the k nearest neighbors, where k is a pre-determined number. One of these k nearest neighbors is then selected at random, x, to compare again with i and the synthetic entry is created using the values from both of these entries. Any feature values that i and x have in common become values in the new synthetic entry. For the features where i and x have different values, one of the values is randomly selected between the two entries to be used in the synthetic entry. This process is repeated so that every entry in the

minority class is used as i, creating a synthetic dataset the same size as the minority class. Combining this with the original minority class effectively doubles the size of the minority class. If more sampling is needed, multiple k neighbors can be selected for each i to create extra synthetic entries, increasing the size of the resulting dataset.

An example of how i and x are compared to create the synthetic entry is illustrated in Table 2.1. Since i and x both have value A for feature 1, the synthetic entry also has value A. On the other hand, i and x have different values for Feature 2. One of the two is randomly chosen and given to the synthetic entry; in this case it gets value B. The same situation occurs again for Feature 3. Feature 4 again has the same value for both i and x, so that value is passed on to the synthetic entry.

Table 2.1: Example of SMOTE-N

	Feature 1	Feature 2	Feature 3	Feature 4
Entry i	A	C	F	G
Entry x	A	B	E	G
Synthetic Entry	A	B	F	G

It is vital that SMOTE is only applied to the training dataset to ensure that model results are not being validated on synthetic data. In this paper, each model was fit to the data using the original training set and a SMOTE version of the training set, and then evaluated using the same unaltered validation set. This is done to understand whether SMOTE improves the performance of the various models or not. Further explanation of this process can be found in section 4.2.

It should also be noted that after applying SMOTE to a dataset, the error function of any classifiers fit to the data should have uniform weighting. Since the data is now artificially balanced, further weighting in the error function would result in a biased classifier.

SMOTE was performed in this paper with the DMwR package in R [3].

2.2 Logistic Regression

One of the most basic tools for predicting a binary outcome is logistic regression. In this paper, logistic regression is used as a baseline to compare whether other specified methods have better results than one of the most straightforward options.

At a high level, logistic regression is a modified form of linear regression that predicts a binary outcome. In this case, it is used to predict whether a lead will become a funded loan or not. A detailed explanation of how logistic regression works can be found in Appendix A.

Logistic Regression is a commonly used modeling technique for predicting binary outcomes and is the first of three modeling methods used in this paper. The results of the logistic regression model will be compared with the results of the other methods in order to understand whether they perform any better on the rare event data. Throughout the rest of this paper, the logistic regression model will be referred to as the Logit model.

In this analysis, Logistic Regression is performed through use of the base stats package in R [4].

2.3 Rare Event Logistic Regression

While logistic regression is a strong tool for predicting a binary output, it has several drawbacks. One of these is that using the logit method with heavily imbalanced data can lead to strong biases in the coefficient estimates. While bias will always be present in coefficient estimates, data with rarely occurring positive cases can magnify this bias and result in wildly varying predictions. A solution to combat this bias is proposed by Langche Zeng and Gary King in their 2003 paper titled 'Logistic Regression in Rare Events Data' [5]. This paper discusses the challenge of performing logistic regression on a dataset where a successful event rarely occurs (otherwise referred to as a "rare event").

Zeng and King studied data with rare events such as war, vetoes, and epidemiological infections in order to explain and predict when these cases might occur. They noted that traditional predictive methods (like logistic regression) tend to grossly underestimate the

likelihood of rare events and considered methods for improving predictions on data with rare events. To solve for this, they proposed a method of bias correction in logistic regression to deal with the bias introduced from imbalanced data. This bias estimate is utilized within a logistic regression model to improve the model's predictions on rare event data. The bias adjustment they proposed is as follows:

$$\text{bias}(\hat{\beta}) = (\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{W}\xi \tag{2.1}$$

Where $\xi_i = 0.5Q_{ii}[(1 + w_1)\hat{\pi}_i - w_1]$, $\hat{\pi}_i$ is the probability assigned to prediction i by the model, Q_{ii} are the diagonal elements of $\mathbf{Q} = \mathbf{X}(\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}'$, and $\mathbf{W}$ here is the same $\mathbf{W}$ from equation A.10, which is a diagonal matrix of the derivatives P_i'. Computing this expression involves running a weighted least-squares regression with $\mathbf{X}$ as the "explanatory variables", ξ as the "dependent variable", and $\mathbf{W}$ as the weight. This resulting bias adjustment is then subtracted from the coefficient estimate $\hat{\beta}$ so that $\tilde{\beta} = \hat{\beta} - \text{bias}(\hat{\beta})$. Once computed, $\tilde{\beta}$ then represents the bias adjusted coefficient estimate and is used in the model equation as follows:

$$\Pr(Y_0 = 1 \mid \tilde{\beta}) = \frac{1}{1 + \exp\left(-x_0\tilde{\beta}\right)} \tag{2.2}$$

Which corresponds with the logistic regression model found in equation A.3.

Zeng and King tested this bias adjusted logistic regression model against traditional logistic regression, as well as models with prior correction and other weighting techniques to deal with imbalanced data. Their bias adjusted logistic regression model showed improvement over all of these other methods, especially when event occurs in less than 5% of the data and when there are only a few thousand observations. This method of logistic regression with the above bias adjustment is referred to as Rare Event Logistic Regression or Relogit, for short.

The derivation of this bias adjustment is explained by Zeng and King in their paper, but it will not be repeated here. For the sake of this paper, it sufficeth to say that this rare event

bias adjustment has been shown to improve the predictive power of logistic regression with rare event data.

In the lead data, funded loans occur in only 0.55% of entries, clearly classifying this as a "rare event". However, the sample size of 494,025 is much larger than the recommended sample size of a few thousand and may minimize the effects of Relogit. The results of Relogit will be compared specifically with that of traditional logistic regression to understand whether the bias adjustment has a noticeable effect.

In this analysis, Relogit is performed through the use of the 'Zelig' package in R [6].

2.4 Gradient Boosted Decision Trees - CatBoost

Departing from the previous two modeling methods based in logistic regression, this final proposed method involves the use of gradient boosted decision trees, specifically a methodology called CatBoost (short for Category Boost). First, a decision tree is a machine learning method used in both regression and classification. Decision trees consist of three main parts: a root node, a decision node, and a terminal node. These three parts mimic a decision-making process that passes through several different deciding criteria and results in a final output. Decision trees are created by choosing certain features that will be used for decision nodes and determining where to make a split within each chosen feature. As a tree grows larger, the splitting decisions become more complex and oftentimes result in overfitting to the training data. This overfitting can be mitigated by fitting smaller trees, which comes at the cost of higher bias.

While a single decision tree can be limited in its predictive power, multiple decision trees can be combined to improve results in what is called an ensemble method. There are many different ensemble techniques, but this paper will focus on boosting, which is a weighted sum of a collection of weak classifiers. Combining several weak classifiers helps to reduce the bias within each individual weak classifier, thus resulting in a strong learner that performs better. The high level model for boosting is as follows:

$$s_l(.) = \sum_{l=1}^{N} c_l \times w_l(.) \tag{2.3}$$

Where c_l is the weight for the weak classifier, w_l represents each weak classifier, N is the number of iterations, and $S_L(.)$ is the resulting strong learner at each iteration.

Boosting begins with a base model, in this case a decision tree. As previously discussed, decision trees tend to have high bias but low variance. To ensure that the decision trees are well-suited for ensembling, the base model for boosting will be a shallow decision tree with few decision nodes. In boosting, a series of classifiers is sequentially trained and build off each other to improve the final result. After training the first weak classifier (or decision tree), the results are passed on to training the next classifier so that it gives more weight to the entries that were previously misclassified. This allows each consecutive classifier to improve upon the errors from the previous iteration. The end result is a strong learner with lower bias.

Gradient boosting is a specific method of boosting that changes the data on which each successive decision tree is fit. The same model for boosting applies, but gradient boosting uses gradient descent to improve the fit on each weak learner. After the first learner is fit to the data, difference between the model predictions and the actual data is calculated for each entry in the training set. These values are known as pseudo-residuals and are attached to each entry in the training set. The next weak learner is then trained using the pseudo-residuals as the response variable in lieu of the actual training set. This continues for a specified number of iterations and results in a strong learner that mitigates the high bias of the individual decision trees. Gradient boosting can be written as follows:

$$s_l(.) = \begin{cases} c_l \times w_l(.) & \text{when } l = 1 \\ s_{l-1}(.) + c_l \times -\nabla_{s_l-1} E_{s_{l-1}}(.) & \text{when } L \geq l > 1 \end{cases} \tag{2.4}$$

Where s_l represents the current ensemble learner, s_{l-1} represents the previous iteration of the ensemble learner, c_l is the coefficient representing the weight of the current weak classifier, w_l is the weak learner fit in the first iteration only, $E_{s_{l-1}}$ is the weak learner fit to

the pseudo-residuals of the previous iteration, $-\nabla_{s_l-1}$ represents the negative gradient of the fitting error with respect to the ensemble model from the previous iteration, and l signifies the current iteration step. This is repeated L times and the L weak learners are aggregated to build the strong ensemble learner.

The steps in fitting a gradient boosted tree are as follows:

1. Fit a weak learner to the data, represented by w_l in equation 2.4.
2. Compute the value of the weight for this weak learner. This is represented by c_l in the first step of equation 2.4.
3. Multiply the weak learner by the weight computed in step 2 ($c_l \times w_l(.)$). This denotes how much this weak learner should contribute to the ensemble model.
4. Use the predictions of this first weak learner to calculate the pseudo-residuals for each entry in the data. This completes the first iteration of the gradient boosting process.
5. Use the pseudo-residuals to fit a weak learner with the negative gradient estimated from the previous strong learner (s_{l-1}). This is represented by $-\nabla_{s_l-1}E_{s_{l-1}}(.)$ in equation 2.4.
6. Compute the value of the weight for this weak learner. This is represented by c_l in the second part of equation 2.4.
7. Multiply the weak learner by the weight computed in step 6 ($c_l \times -\nabla_{s_l-1}E_{s_{l-1}}(.)$). This denotes how much this weak learner should contribute to the ensemble model.
8. Add this weighted learner to the previous ensemble model, represented by $s_{l-1}(.)$) in equation 2.4. This results in the updated ensemble model for the current iteration, $s_l(.)$.
9. Use the current ensemble model to calculate the new pseudo-residuals for each entry in the data.
10. Repeat steps 5 through 9 for L iterations to build the ensemble model.

In 2017, Russian company Yandex launched an open source machine learning library based on gradient boosting, named CatBoost [7]. This new methodology brings two main features to traditional gradient boosting [8]:

1. Ordered Target Statistic - A new method for processing categorical features.

2. Ordered Boosting – A permutation-drive alternative to traditional boosting.

The creators of CatBoost created these two methods to mitigate a kind of "target leakage" that is present in existing implementations of gradient boosting algorithms.

First, a common approach for dealing with categorical variables in gradient boosting is the Target Statistic or target mean encoding. This involves encoding each categorical feature with the estimate of the expected response for that category. However, this process can easily lead to overfitting and bias in the resulting data. CatBoost improves upon this Target Statistic by introducing an artificial concept of "time" into the data, called the Ordered Target Statistic. To do this, a random permutation of the training examples is selected. Then, for each example, the Target Statistic is calculated using the current example and all of the preceding samples in this artificial "timeline" of data. This means that the values of the Ordered Target Statistic for each example rely on the observed history, which works to eliminate the "leakage" present in calculating the traditional Target Statistic. For those variables with only two unique values, One Hot Encoding is used.

The second main feature of CatBoost is Ordered Boosting. The problem with traditional gradient boosting is that the gradients at each step are calculated with the same data points that the current model was built on. This reuse of data results in overfitting because of biased gradients. To combat this, CatBoost creates separate models for each entry in the training set that are not updated with a gradient estimate for that example. These separate models are then used to get unbiased estimates in the gradient step.

Ordered Boosting begins with the same steps as traditional gradient boosting. The difference lies in how the gradient value is calculated for each training sample. Begin with F_i, the resulting model after building i trees, and $g^i(\mathbf{X}_k Y_k)$ as the gradient value on the

k-th training sample. To obtain an unbiased gradient value, a separate model M_k is trained that is not updated with a gradient estimate for k. The resulting model can then be used to estimate the gradient on $\mathbf{X}_k$, which is then used to score the resulting tree. Thus, the gradient in each step is not subject to bias, improving the results of the resulting gradient boosted model.

With these two features, CatBoost is a powerful tool for fitting Gradient Boosted Decision Trees. In this paper, the CatBoost package in R [9] was used to fit Gradient Boosted Trees to the dataset.

All of these methods will be used in this paper to create a model that can best predict which leads are most likely to become funded loans. The results of each model type will be compared to the other two when trained on the original training set and when trained on the SMOTE training set.

CHAPTER 3

Feature Exploration and Selection

Each entry in this dataset contains thirteen explanatory variables and one response variable. The names of each of these variables, as well as a brief description, is given below:

Purchase Hour - *Categorical* - The hour of the day in which the lead was purchased.
Loan Value - *Numeric* - The balance of the mortgage to be refinanced.
Credit Grade - *Categorical* - The self-reported credit grade of the consumer.
Loan-to-Value Ratio - *Categorical* - The ratio of loan amount to home value.
Add Cash Amount - *Numeric* - The amount of additional cash that the consumer would like to take out with the refinance process.
Property Description - *Categorical* - A short description of the property type on which the existing mortgage is backed.
1st Mortgage Interest Rate - *Numeric* - The interest rate on the existing mortgage.
2nd Mortgage - *Binary* - Whether there exists a second mortgage on the property.
Number of Mortgage Lates - *Categorical* - The number of late mortgage payments made in the last 3 years.
VA Status - *Binary* - Whether the consumer qualifies for a government-backed U.S. Department of Veteran's Affairs (VA) loan.
FHA Eligible - *Binary* - Whether the consumer qualifies for a government-backed Federal Housing Administration (FHA) loan.
Loan Type - *Categorical* - Describes the type of interest rate on the existing mortgage; whether it is fixed or adjustable.
Income - *Numeric* - The estimated median income of the zip code in which the consumer's

property resides.

Funded - *Binary* - This is a binary flag that indicates whether a lead became a funded loan or not. This is the response variable for the data.

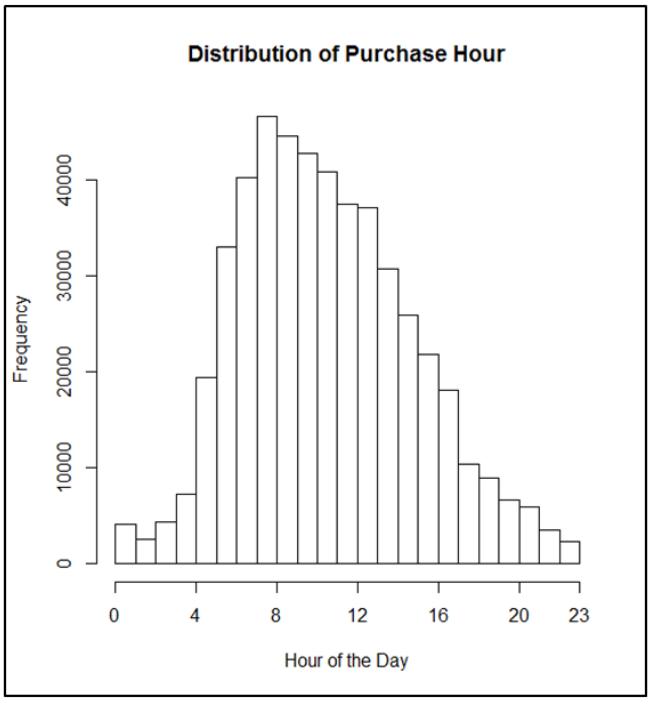
The following section describes each feature in detail, as well as the distribution of each, and any potential considerations for feature engineering.

3.1 Exploratory Data Analysis

Purchase Hour

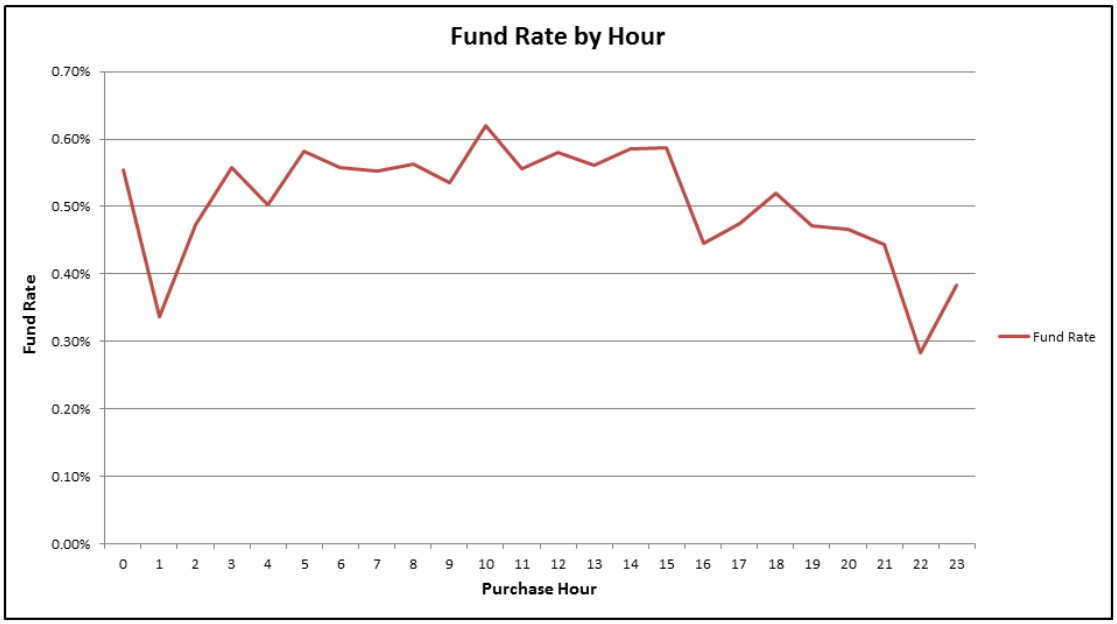
The first feature is purchase hour, the hour of day in which the lead was purchased by the lender. This feature is a categorical variable and figure 3.1 shows the distribution of values. The vast majority of leads are purchased between 5 am and 6 pm, following common business hours during the day.

Figure 3.1: Histogram showing the distribution of leads by hour.



Leads purchased in the night between 9 pm and 8 am (local time) cannot be immediately contacted due to the Telephone Consumer Protection Act of 1991 [10], so these leads are held until the proper hours when the lenders can legally try to contact the consumers. Due to this delay, it is possible that leads may perform differently depending on when they are purchased. Table 3.1 shows the number of leads purchased in each hour, as well as the fund rate from all leads purchased within that hour and figure 3.2 plots the fund rate by hour.

Figure 3.2: Plot of the fund rate by hour.



The fund rate appears to be largely consistent throughout most of the day until hour 16 when it begins to drop off fairly significantly. Even though the sampling decreases in the later hours, this shift in fund rate could be a helpful predictor. Unfortunately, most of the hour-to-hour variation in fund rate just adds noise to the data that might limit the effectiveness of Purchase Hour as a feature. For example, even though hour 8 and 9 have different fund rates (0.57% and 0.54%, respectively), this difference is unlikely to provide any really useful information. This feature will be more useful in identifying the general trend in fund rate by hour, rather than the individual differences between each hour.

To that end, the Purchase Hour feature can be grouped into different time periods throughout the day. This groups similar hours of the day and works to reveal the main

Table 3.1: Lead count and fund rates for leads purchased during each hour of the day

Hour	**Leads**	**Funded Loans**	**%Fund**
0	1,973	11	0.56%
1	2,071	7	0.34%
2	2,521	12	0.48%
3	4,275	24	0.56%
4	7,136	36	0.50%
5	19,306	113	0.59%
6	32,790	184	0.56%
7	39,981	222	0.56%
8	46,343	262	0.57%
9	44,344	239	0.54%
10	42,432	265	0.62%
11	40,621	227	0.56%
12	37,162	217	0.58%
13	36,866	208	0.56%
14	30,544	180	0.59%
15	25,747	152	0.59%
16	21,713	97	0.45%
17	17,992	86	0.48%
18	10,344	54	0.52%
19	8,863	42	0.47%
20	6,616	31	0.47%
21	5,826	26	0.45%
22	3,510	10	0.28%
23	2,335	9	0.39%

trend in the fund rate. Consider the grouping presented in table 3.2 which splits the day into four different groups of six hours. The lead count is not evenly distributed in this grouping, but this makes it clear that leads purchased in the evening are not as likely to fund as those purchased during the day.

Table 3.2: Lead count and fund rate for the groupings of purchase hour.

Hour	**Purchase Period**	**Leads**	**Funded Loans**	**%Fund**
0 - 5 am	Night	37,485	203	0.54%
6 am - 12 pm	Morning	247,910	1,399	0.56%
1 - 6 pm	Afternoon	170,964	940	0.55%
7 - 11 pm	Evening	37,666	172	0.46%
Grand Total		**494,025**	**2,714**	**0.55%**

Loan Value

The next feature is the total value of the mortgage loan to be refinanced. Higher loan values net more profit for the lender but also assume a higher risk of defaulting, which may lead to lower fund rates on larger loans. It is also worth noting that there is more competition on higher loan values, often leading to reduced fund rates since only one lender can ultimately close the loan.

Table 3.3 shows a summary of how the loan values are distributed and figure 3.3 shows a histogram of the loan value distribution. It is readily apparent that the loan values are heavily right-skewed and there are a few potential outliers on the high end.

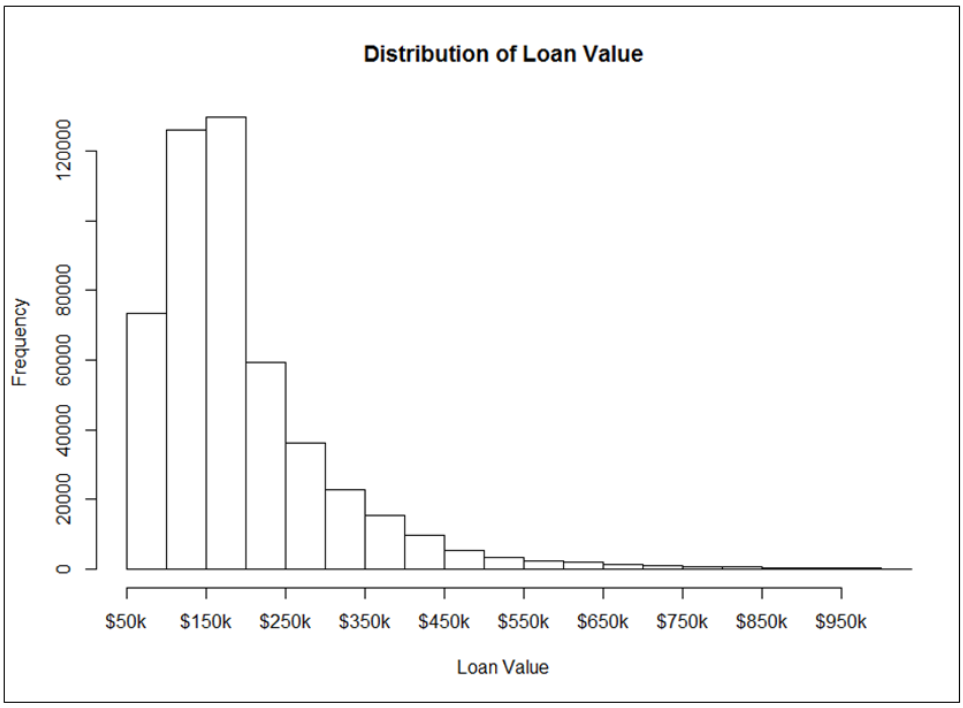
This feature may work as a numeric variable, but also may benefit from grouping the loan values into groups and treating it as a categorical variable. Grouping the loan values would also mitigate the effects of the outliers present in this feature.

Consider the grouping in table 3.4, splitting the loan values into four buckets. This

Table 3.3: Six number summary for the Loan Value variable

Min.	1st Qrt.	Median	Mean	3rd Qrt.	Max.
$50,000	$130,000	$160,000	$206,081	$235,000	$2,700,000

Figure 3.3: Histogram of values for Loan Value in the lead dataset.



grouping clearly reveals the decreasing trend in fund rate as the loan value increases. Unfortunately, the larger loan amounts have sparse sampling, which may lead to bias in the model. The strength of this grouping is in the split between $50k-$124k and $125k-$453k. These two buckets contain the vast majority of the leads and there is a clear difference in fund rate between loans in these ranges.

The feature selection section will consider the performance of this feature as both a numeric and categorical variable.

Table 3.4: Lead count and fund rate for the proposed Loan Value groupings

Loan Value	Leads	Funded Loans	%Fund
$50k - 124k	116,640	933	0.80%
$125k - 453k	356,194	1,732	0.49%
$454k - 649k	13,209	37	0.28%
$650k+	7,982	12	0.15%

Credit Grade

A consumer's credit score acts as a measure of risk-assessment and plays a major role in whether a loan funds or not. Credit scores range from 300 to 900 and are split into four main categories: Excellent, Good, Fair, and Poor. The better a consumer's credit score, the less likely they are to default on their loan, which makes them a better candidate for refinancing.

The leads in this dataset don't contain an actual FICO or credit score. Instead, the consumer is asked to give their self-assessed credit grade. The online forms [11] that generated the leads in this dataset provide guidance on estimating their credit rating as seen in table 3.5. This can sometimes result in consumers over-reporting their credit rating in an attempt to get a better offer from a lender. However, once a lender contacts a consumer, one of the first steps in the funding process is to run an official credit report on that consumer. If this official credit score is too low, the consumer is automatically disqualified from the refinancing process. Fortunately, as lender data has confirmed, most people tend to report their credit grade accurately enough for use in the funding process.

Table 3.6 shows the distribution of credit grades, as well as the fund rate for each. First, there are no poor credit leads because the buyers are not interested in purchasing those type of leads. Interestingly, Good credit appears to have a slightly higher fund rate than Excellent credit. One potential explanation for this is that Excellent credit leads tend to

Table 3.5: A common example of Credit Grade guidance found on online mortgage forms.

Credit Grade	FICO Range
Excellent	Above 680
Good	639 - 679
Fair	560 - 599
Poor	Below 559

have more competition, which leads to lower fund rates, on average. Fair credit leads appear underrepresented in this data, but clearly show a lower fund rate than higher credit grades, making this feature potentially valuable in the model.

Table 3.6: Lead count and fund rate for each Credit Grade.

Credit Grade	Leads	Funded Loans	%Fund
Excellent	191,259	993	0.52%
Good	255,690	1,521	0.59%
Fair	47,076	200	0.43%

Loan to Value Ratio (LTV)

The Loan to Value Ratio is simply the refinance loan amount divided by the property value. This feature is pre-grouped in the data to represent the maximum LTV value within a range of 5. For example, any LTV between 96-100 is labeled as 100, any LTV between 91-95 is labeled 95, and so on down to 70, which includes any LTV below 70. Figure 3.4 shows a histogram of the distribution of LTV ratios in the data and table 3.7 shows the fund rate for each LTV.

The majority of leads are below 70 LTV and very few exceed 100 LTV. It would be reasonable to expect lower LTVs to have a higher propensity to fund, although that doesn't

Figure 3.4: Histogram of values for LTV in the lead dataset.

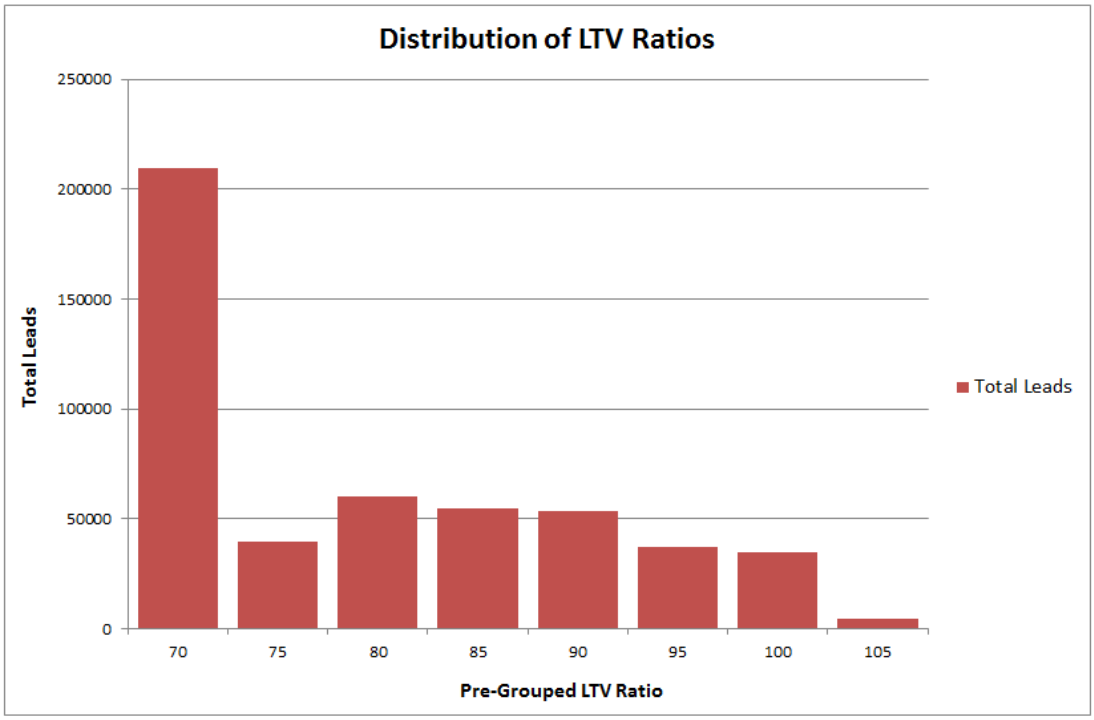


Table 3.7: Lead count and fund rate for LTV Ratio.

LTV	Leads	Funded Loans	%Fund
70	209,537	1,142	0.55%
75	39,784	238	0.60%
80	60,108	293	0.49%
85	54,960	293	0.53%
90	53,452	315	0.59%
95	36,942	212	0.57%
100	34,594	184	0.53%
105	4,647	37	0.80%

appear to be the case here. Surprisingly, the highest fund rate is on 105 LTV, although this may be a result of low sampling and decreased competition on those leads.

This feature could be used as-is, but might benefit from further grouping into fewer categories. Table 3.8 shows a suggested grouping and the corresponding fund rates. While this grouping still has imbalanced sampling, it more clearly indicates the differences in fund rates between the LTV values.

Table 3.8: Lead count and fund rate for the newly proposed grouping of LTV Ratios.

LTV	**Leads**	**Funded Loans**	**%Fund**
70 - 75	249,321	1,380	0.55%
76 - 85	115,068	586	0.51%
86 - 95	90,394	527	0.58%
96+	39,242	221	0.56%

Add Cash

When refinancing a mortgage, there is an option to take out a higher loan amount and receive the extra amount in cash from the lender. This extra amount is backed by equity in the property. Many consumers refinance their loan in order to take cash out, while others do so to re-negotiate their interest rate or payment terms and don't increase the value of the loan.

The distribution of add cash values is shown in table 3.9. The median of 0 indicates that at least 50% of these leads are not looking to take cash out with their refinance. Figure 3.5 shows the histogram of the distribution of add cash values above 0 with high outliers removed.

Table 3.9: Six number summary for the Add Cash variable.

Min.	**1st Qrt.**	**Median**	**Mean**	**3rd Qrt.**	**Max.**
0	0	0	$8,791	$5,000	$2,000,000

Similar to Loan Value, this feature is also heavily right-skewed and further complicated by the presence of $0 indicating no added cash. Due to these challenges, this feature is likely to struggle as a numeric feature and would benefit from grouped categorical values.

Table 3.10 proposes grouping the add cash variable into three groups. Leads with low add cash values have similar fund rates as leads with $0 add cash, leading to a natural grouping. The fund rate tends to increase on leads with higher add cash amounts, potentially indicating that consumers in need of large sums of cash are more motivated to complete the refinancing process. However, this trend flattens out for add cash values above $10k, leading to the grouping of all add cash values of $10k and above.

Feature selection will test both the numeric and grouped categorical options for the add cash feature.

Figure 3.5: Histogram of values for the Add Cash variable.

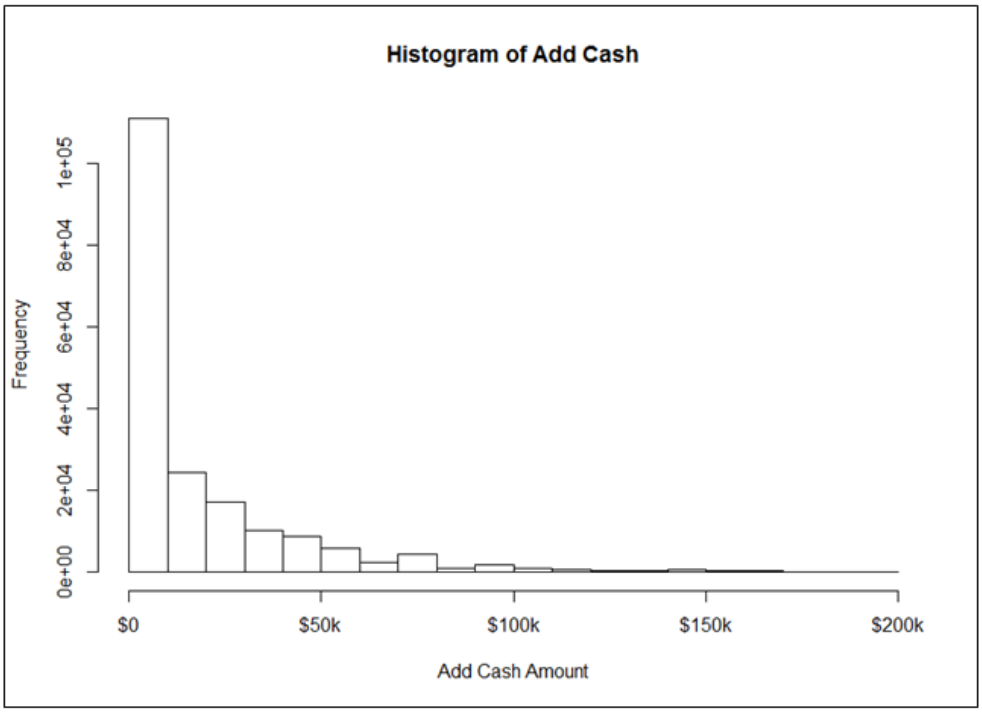


Table 3.10: Lead count and fund rate of proposed grouping for Add Cash.

Add Cash Amt.	Leads	Funded Loans	%Fund
$0 - 5k	302,332	1,376	0.46%
$5k - 9k	92,885	529	0.57%
$10k+	98,808	809	0.82%

Property Description

The Property Description is a straightforward variable that describes the type of house backing the mortgage. By far, the majority of homes are Single Family residences, which brings an unfortunate imbalance to the data distribution. These types of properties are also the most likely to fund, as indicated in table 3.11. No further engineering appears necessary for this variable.

Table 3.11: Lead count and fund rate of proposed grouping for Add Cash.

Property Type	**Leads**	**Funded Loans**	**%Fund**
Multi Family	19,592	67	0.34%
Single Family	458,551	2,582	0.56%
Townhome	15,882	65	0.41%

1st Mortgage Interest Rate

Each lead contains the interest rate on the current mortgage, self-reported by the consumer. Reducing the interest rate on a mortgage is a common reason for refinancing a loan, whether or not any additional cash is taken out with it. A reduced interest rate generally means lower monthly payments, which is a very strong motivator for any consumer. Interest rates tend to vary between 2% and 8%, with a few outliers above 8%, as shown in the histogram in figure 3.6.

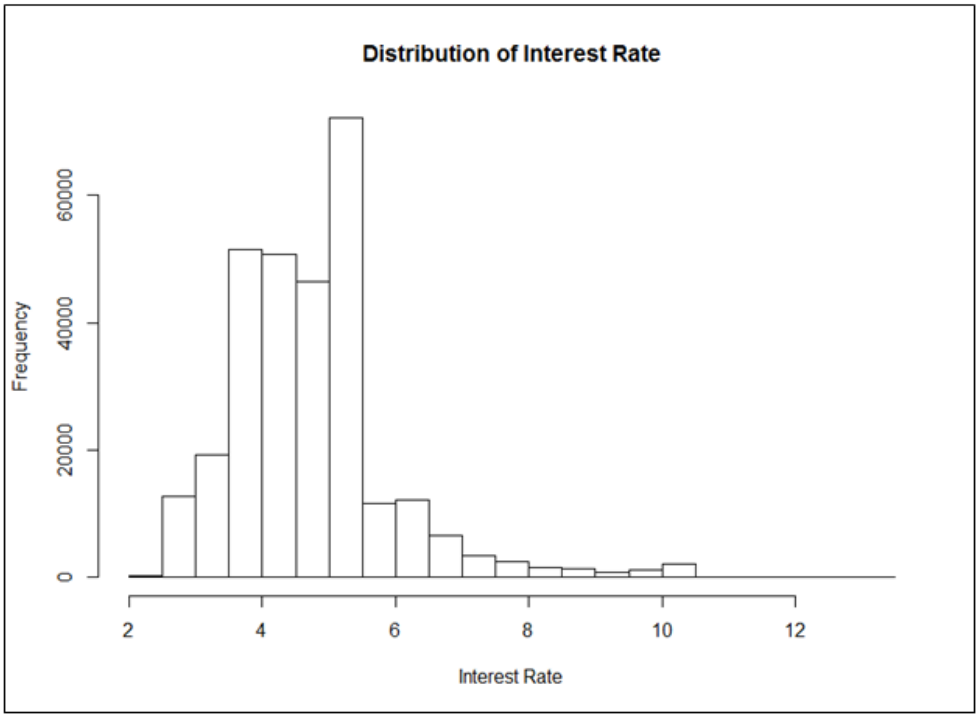
This feature is also right-skewed and, as seen in table 3.12, has a large number of missing values. Similar to the problem discussed with Property Description, a blank interest rate doesn't possess any meaningful real-world interpretation. And excluding the rows with NA interest rate would be detrimental since that would remove almost 40% of the values in the dataset. It may be reasonable to assume the median interest rate for these NA values rather than dropping these values or this feature entirely.

Table 3.12: Six number summary for the 1st Mortgage Interest Rate variable.

Min.	**1st Qrt.**	**Median**	**Mean**	**3rd Qrt.**	**Max.**	NA
2	4	4.75	4.84	5.25	13.5	197,793

This solution is more feasible if the values of this feature are grouped into smaller buckets, as proposed in table 3.13. The majority of leads fall between 2% and 5.9%, which is where

Figure 3.6: Histogram of values for the Add Cash variable.



the NA leads would also fall. The fund rate does not vary much between those values since consumers who already have lower interest rates are not likely to get a much lower rate. As the interest rate climbs, so does the fund rate (as well as that consumer's desire to lower their monthly payments). Even though this grouping is somewhat imbalanced, the increasing trend in fund rate indicates a useful variable.

Table 3.13: Lead count and fund rate of proposed grouping for Interest Rate.

Property Type	**Leads**	**Funded Loans**	**%Fund**
2 - 5.9%	455,361	2,448	0.54%
6 - 6.9%	22,133	133	0.65%
7%+	16,531	123	0.74%

2nd Mortgage

This is a binary variable that indicates whether the consumer currently has a second mortgage on their residence. As table 3.14 shows, this feature is often blank in the data; almost 25% of leads are missing a value for this feature.

Table 3.14: Lead count and fund rate of for the values of the 2nd Mortgage variable.

2nd Mortgage	Leads	Funded Loans	%Fund
Yes	29,046	161	0.55%
No	342,894	1,858	0.54%
(Blank)	122,085	695	0.57%

Interestingly, the blank fields have a slightly higher fund rate than both the "Yes" and "No" responses. However, it does not make practical sense to leave these fields blank because there is a true answer for this field and leaving that information out of the lead should not affect the lead's propensity to become a funded loan.

Unfortunately, there isn't a great solution for re-coding these blank values. It would not be reasonable to simply encode all of the missing values as "Yes" or "No" because that would require making too general of an assumption about the data. As a result, this variable will not be utilized in fitting a model.

Number of Mortgage Lates

The Number of Mortgage Lates indicates the number of late payments made on their mortgage in the last year, as self-reported by the consumer. Consumers with late payments represent a higher risk for the lender, which reduces the likelihood of refinancing the loan. This trend is clearly represented in table 3.15.

Table 3.15: Lead count and fund rate of proposed grouping for Number of Mortgage Lates.

Late Payments	Leads	Funded Loans	%Fund
None	437,994	2,505	0.57%
One	25,816	112	0.43%
Two or More	30,215	97	0.32%

VA Status

The United States Department of Veterans Affairs (VA) is federal agency that provides a large number of services to eligible military veterans and current military service members. One of these services includes a special government-backed loan called a VA Loan. These loans are partially backed by the Department of Veterans Affairs, which allows the lender to give more favorable terms to the consumer without taking on higher risk. For this reason, VA loans are more likely to fund, as displayed in table 3.16.

Table 3.16: Lead count and fund rate of proposed grouping for VA Status.

VA Status	Leads	Funded Loans	%Fund
No	336,145	1,841	0.55%
Yes	127,393	764	0.60%
(Blank)	30,487	109	0.36%

Missing values are present once again in this feature. It doesn't seem sensible to assume that unknown consumers have served in the military and the most common response is "No", which naturally leads to re-coding the missing values as "No". It seems odd that the missing values have such a lower fund rate than the "No" or "Yes" responses, but it doesn't seem prudent to remove them entirely.

FHA Status

The Federal Housing Administration (FHA) is another federal agency that provides government-backed mortgage insurance on loans that meet certain requirements. A mortgage with FHA insurance protects the lender from potential losses and allows the lender to give more favorable terms to the consumer. The specific requirements to take out an FHA loan are as follows:

1. Must be able to document income.
2. Must maintain a debt-to-income ratio below 43%
3. The loan amount must be below a certain amount, depending on the residence type and where it is located.

When a lead is submitted, the FHA Status does not indicate whether the current mortgage is an FHA loan or not, but rather, whether it meets the requirements to become an FHA loan. Table 3.17 shows the distribution of FHA eligibility among leads in the data.

Table 3.17: Lead count and fund rate of for the values of the 2nd Mortgage variable.

FHA Status	**Leads**	**Funded Loans**	**%Fund**
True	149,826	995	0.66%
False	344,199	1,719	0.50%

About 30% of these leads are eligible to become an FHA loan, as expected, given the rather stringent requirements. There is a very clear delta in performance, as leads that are FHA eligible tend to fund more often than the leads not eligible for FHA status. Fortunately, this feature is fairly straightforward and doesn't require any further engineering.

Loan Type

When a loan is issued, there are two different types of interest rates that could be applied to the balance. The first is a fixed interest rate, which is determined in the qualification process and does not change throughout the life of the loan. The other is an adjustable interest rate, which is steady for an initial period of time and then becomes a variable rate that changes at a specific cadence. Adjustable interest rates are often initially lower than fixed rates, but can increase drastically once the initial period is over.

The distribution of loan types can be seen in table 3.18. Surprisingly, loans with a fixed interest rate tend to fund more often than loans with adjustable rates. The distribution is skewed toward fixed rates as only 6% of leads in the data are listed as adjustable. Since this feature only has two options, there isn't really any engineering to do here.

Table 3.18: Lead count and fund rate of for the values of Loan Type.

Loan Type	**Leads**	**Funded Loans**	**%Fund**
Adjustable	29,854	124	0.41%
Fixed	461,457	2,590	0.56%

Income

The income value is not a value reported by the consumer. Instead, it is derived from the estimated median yearly income of the reported zip code. While this is by no means an exact measurement, it is a relative indicator of a consumer's affluence. Figure 3.7 shows the distribution of incomes listed in the data and table 3.19 shows a summary of the data. As expected, the data is right-skewed, but there doesn't seem to be an abundance of outliers here.

One might expect that higher incomes would be more likely to fund. However, that does not seem to be the case here, as demonstrated in figure 3.8. It actually appears that the fund rate drops as income increases past $90k. A possible explanation for this is that consumers

Figure 3.7: Histogram of values for the Income variable.

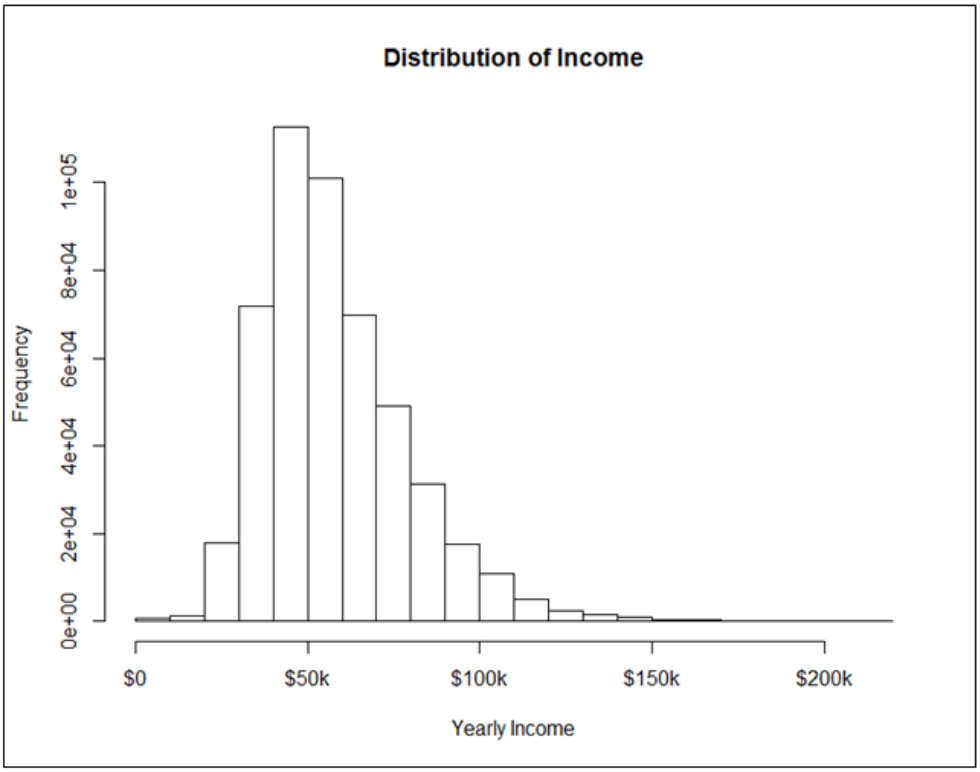


Table 3.19: Six number summary for the Income variable.

Min.	1st Qrt.	Median	Mean	3rd Qrt.	Max.
$33	$42,992	$53,805	$57,970	$69,348	$219,554

with higher income are more likely to shop around when considering refinancing their loan since they are not hard-pressed on monthly payments or need to take out cash immediately.

Due to this trend, it may be beneficial to group these values and treat them as a categorical variable instead of numeric. The proposed grouping is shown in table 3.20. This grouping reduces the feature down to three categories, which each capture different trends in the data. It is important to separate $70-99k from Below $70k because $70k marks the beginning of the declining fund rate (as seen in figure 3.8). Similarly, $100k+ incomes must be separated from $70-99k because the fund rate is drastically reduced for incomes above $99k.

Figure 3.8: Plot of the average fund rate by income.

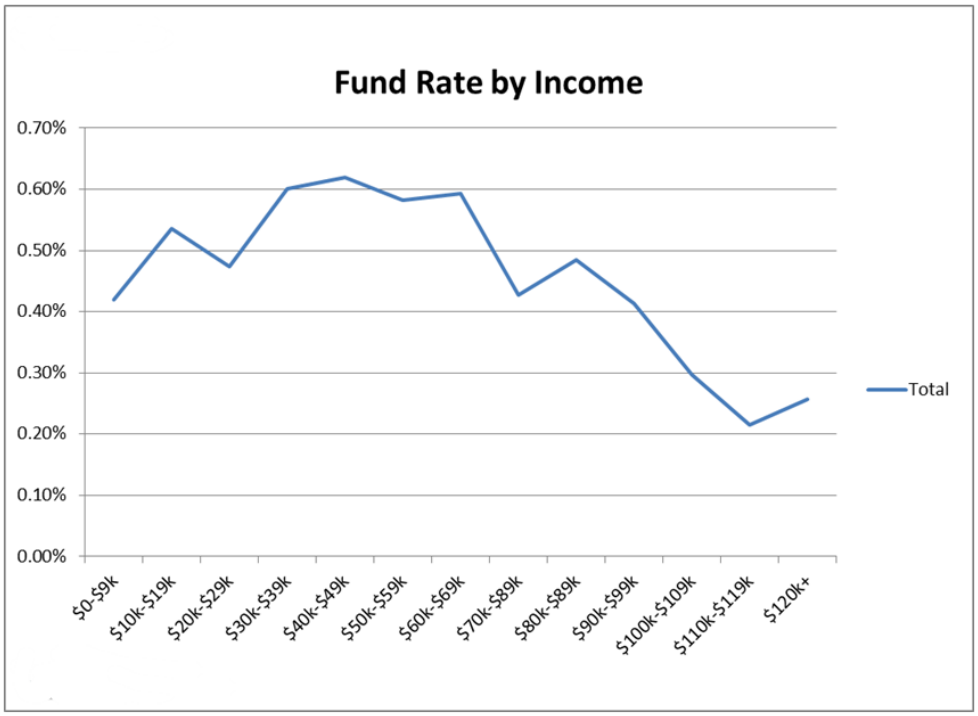


Table 3.20: Lead count and fund rate of proposed grouping for Income.

Income	Leads	Funded Loans	%Fund
Below $70k	374,579	2,223	0.59%
$70 - 99k	97,731	433	0.44%
$100k+	21,715	58	0.27%

3.2 Feature Selection

Based on the exploratory data analysis, all of the variables appear viable for use in a predictive model, with the exception of 2nd Mortgage. The following section will determine which of these remaining features to select for the model through the use of two methods: The Chi-Squared Test and the Information Value calculation.

3.2.1 Chi-Squared Test

Feature selection begins with Pearson's Chi-Squared Test to determine the significance of the categorical and binary variables. There are eight categorical or binary features and the remaining four numeric features all have proposed categorical groupings that can be tested here as well. The results of Pearson's Chi-Squared Test don't necessarily determine which features are best suited for the model, but it will be used to select the variables with the strongest correlation and to remove any variables that show very weak correlation.

The Chi-Squared test was run on all 8 categorical and binary features as well as the categorical grouping of the 4 numeric features. Figure 3.21 shows the p-value and Chi-Squared statistic for each variable. Features with a statistically significant p-value (in this case, below 0.05) are considered to have strong correlations with funding, while the non-statistically significant p-values are unlikely to have any detectable influence on funding.

Based on the results (shown in table 3.21), there is evidence to suggest that LTV Ratio and Purchase Period will not add any meaningful information to the model. It should be noted that even though LTV Ratio and Purchase Period also have P-Values above 0.05, their P-Values are still relatively low, which warrants further testing to determine whether they should be included in the final model.

The information in table 3.21 is not particularly informative because it shows every variable as statistically significant. There are some variables that appear to be more significant than others (specifically Loan Value and Add Cash amount, both with large values for the Chi-Squared Statistic), but this alone cannot determine the best features for the model.

3.2.2 Information Value

The next step in this feature selection process is to calculate the Information Value for each categorical feature. This is used to determine how important each feature is in explaining the response variable. Information Value is calculated using the following equation for the values in each variable:

Table 3.21: Results of the Chi-Squared test on each of the 13 variables.

Feature Name	**Chi-Squared Statistic**	**P-Value**
Purchase Period	6.98	0.0724
Mtg. Two	1.25	0.5346
VA Status	26.47	~0
FHA Eligible	51.55	~0
Loan Value	200.76	~0
Credit Grade	26.15	~0
LTV Ratio	5.47	0.1402
Add Cash Amount	181.08	~0
Property Description	21.63	~0
Mtg. One Interest Rate	16.41	0.0003
Mortgage Lates	39.22	~0
Loan Type	10.5	0.0012
Income	65.22	~0

$$\text{Information Value} = \sum (\%\text{Success} - \%\text{Non-Success}) \times \ln\left(\frac{\%\text{Success}}{\%\text{Non-Success}}\right) \tag{3.1}$$

In this case, %Success represents the total number of funded loans for a single value of any feature divided by the total number of funded loans for all values of that feature. Similarly, %Non-Success represents the total number of non-funded loans for a single value of any feature divided by the total number of non-funded loans for all values of that feature. This is calculated and then summed across all values of a feature to get the Information Value for that feature. Table 3.22 demonstrates this calculation for the Credit Grade feature.

The Information Value calculation for each feature gives an indication of that feature's predictive power. Traditionally, features with strong predictive power will have values of 0.3 or above, medium strength features will have values between 0.1 and 0.3, weak strength features will have values between 0.02 and 0.1, and features below 0.02 and not useful for

Table 3.22: Demonstration of how to calculate the Information Value of the Credit Grade feature.

Credit Grade	**Total Leads**	**Total Funded**	**Total Non-Funded**
Excellent	191,259	993	190,266
Good	255,690	1,521	254,169
Fair	47,076	200	46,876
Total	**494,025**	**2,714**	**491,311**

Credit Grade	**%Success**	**%Non-Success**	**(%Success - %Non-Success)**
Excellent	$\frac{993}{2,714} = 0.366$	$\frac{190,266}{491,311} = 0.387$	-0.021
Good	$\frac{1,521}{2,714} = 0.560$	$\frac{254,169}{491,311} = 0.517$	0.043
Fair	$\frac{200}{2,714} = 0.074$	$\frac{46,876}{491,311} = 0.095$	-0.022

Credit Grade	$\mathbf{ln\frac{\%Success}{\%Non\text{-}Success}}$	**Information Value**
Excellent	-0.057	0.001
Good	0.080	0.003
Fair	-0.258	0.006
Total		**0.010**

prediction. The results of this calculation for each feature are shown in table 3.23.

It is readily apparent that none of the variables in this dataset have Information Values that indicate typically strong predictive power. This is mostly due to the strong imbalance in the data making the variables seem less important. To account for this, the traditional scoring will not be applied to the variables in this dataset. Instead, the Information Value for each variable is used as a guide to assemble five different feature sets that will all be used for each model. These five features sets are listed in table 3.24. The first feature set only includes the three variables that have an Information Value above 0.02. The second feature set has the same variables as feature set 1, in addition to the two variables with the next highest Information Value at 0.018. Feature sets 3 and 4 both add on two more

Table 3.23: Information Value for each feature.

Variable	**Information Value**
Loan Value	0.075
Add Cash Amount	0.060
Income	0.030
FHA Eligible	0.018
Mortgage Lates	0.018
Credit Grade	0.010
Property Description	0.010
Mtg. One Interest Rate	0.005
Loan Type	0.005
VA Status	0.003
Purchase Period	0.003
LTV Ratio	0.002
Mtg. Two	0.000

variables with the two highest Information Values. And finally, feature set 5 contains all of the variables with an Information Value above 0. In feature Set 5, the only missing variable is Mtg Two, which follows from the Chi-Squared test in table 3.21. Each model will be tested on each of the feature sets to find the best performing set of features for each model type.

This feature selection was performed only on the categorical variables and the categorical groupings of the numeric variables. All of the features with numeric values are present in at least one of the proposed feature sets. To test whether the numeric values are better than the categorical groupings, both methods will be tested in the model comparison section.

Table 3.24: List of the five proposed feature sets.

Feature Set 1	**Feature Set 2**	**Feature Set 3**	**Feature Set 4**	**Feature Set 5**
Loan Value	Loan Value	Loan Value	Loan Value	Loan Value
Add Cash Amount	Add Cash Amount	Add Cash Amount	Add Cash Amount	Add Cash Amount
Income	Income	Income	Income	Income
	FHA Eligible	FHA Eligible	FHA Eligible	FHA Eligible
	Mortgage Lates	Mortgage Lates	Mortgage Lates	Mortgage Lates
		Credit Grade	Credit Grade	Credit Grade
		Property Description	Property Description	Property Description
			Mtg. One Interest	Mtg. One Interest
			Loan Type	Loan Type
				VA Status
				Purchase Period
				LTV Ratio

CHAPTER 4

Methods for Model Evaluation

As discussed previously, this dataset contains 494,025 leads, of which, only 0.55% became funded loans. Obviously, this means that 99.45% of the leads did not fund. One of the problems with evaluating models on this dataset is that if any given model is fit to the data and predicts that none of the leads will fund, it is immediately 99.45% accurate. This gross imbalance in the dataset necessitates different methods of evaluating a model's results.

4.1 Performance Metrics

The goal of this model is to extract a set of leads that are most likely to become a funded loan. If the model is capable of selecting a group of leads that has a higher fund rate than the rest of the population, these select leads would be worth more to the buyers. This would allow LeadPoint to sell these leads for a higher price than the current price. With this in mind, the model should maximize the precision of its predictions, which is measured as the percentage of fund predictions that are actually true. This is measuring the true positive rate of the model's predictions.

Furthermore, the precision is only important as it relates to the population's true fund rate. A model with a precision less than the fund rate of the population indicates that the model is failing to find the leads most likely to fund. To measure this, the model needs to be evaluated on how much its precision improves over the population's true fund rate. This is derived as follows:

$$\text{Improvement Metric} = \frac{(\sum \text{Correct Positive Predictions} / \sum \text{Positive Predictions})}{(\sum \text{Population Funded Loans} / \sum \text{Population Lead Count})} \quad (4.1)$$

This Improvement Metric is the most important measurement to evaluate the model because it determines how much value is added to the leads in the predicted set. As the model maximizes the Improvement Metric, it is accurately predicting the high quality leads without over-predicting the number of leads that will fund.

While maximizing Improvement Metric is the main goal of the model, this cannot be the only goal. Otherwise a model could successfully predict a single funded loan and have an Improvement Metric of over 150. Even though that maximizes the Improvement Metric, it is not practically useful to only select 1 funded loan out of 494,025 leads. Thus, it is important to evaluate the model across other additional measurements to ensure that the model is accurately predicting as many leads as possible.

With this in mind, it is also important to consider how many of the true funded loans that the model is able to extract from the data. A model that is correct on 40% of its predictions is not valuable if it is only able to predict 2% of the total funded loans. This measurement is called the Specificity of the model and it is derived as follows:

$$\text{Specificity} = \frac{\sum \text{Correct Positive Predictions}}{\sum \text{Actual Funded Loans}} \quad (4.2)$$

The Specificity is the second most important measurement for evaluating the model because it will measure how effective the model is at extracting the high quality leads and it will also determine the appropriate value of the remaining leads that are not predicted to fund. As the model maximizes the Specificity, it will more accurately extract the highest quality leads.

The final metric to consider is the model's ability to predict which leads are the least likely to fund. It is vital that the model isn't simply guessing that most of the leads will fund because that waters down the value of the predictions. This measurement is called the Sensitivity of the model, and it measures the percentage of true non-funds that the model

is able to accurately predict. This metric is derived as follows:

$$\text{Sensitivity} = \frac{\sum \text{Correct Negative Predictions}}{\sum \text{Population Count of Non-Funded Leads}} \tag{4.3}$$

The Sensitivity is the least important measurement for model evaluation. This is because part of this metric is already captured in the Improvement Metric, although the Sensitivity alone still provides relevant information about the model's performance. As the model maximizes the Sensitivity, it will more accurately weed-out the low quality leads.

4.2 Splitting the Data

Each model will be trained on the same training dataset and evaluated on the same validation dataset. The models will be evaluated by each of these three performance metrics on the validation set and compared to determine which model is the most effective. This will be done for each of the five feature sets.

The data has been split into three different parts for the modeling process. First, the training dataset contains 50% of the total data. Table 4.1 shows the distribution of leads and funded loans within all three splits of the dataset. The validation dataset contains 25% of the total data. The remaining 25% of the data will be used for a test dataset to evaluate the final results of the model with the best performance on the validation dataset.

Table 4.1: Distribution of leads and fund rate between the three splits of the dataset.

Dataset Name:	Training	Validation	Test
Total Leads:	247,013	123,506	123,506
Total Funded Loans:	1,371	679	664
Fund Rate:	0.56%	0.55%	0.54%

4.3 Choosing the Prediction Threshold

When a model is tested on the validation set, it returns a value between 0 and 1 that measures the propensity of that entry to be a funded loan. For each model, a cutoff point must be chosen that indicates how entries are classified. Entries with predicted values above the cutoff will be classified as funded loans and entries with values below the cutoff are classified as non-funding loans. Traditionally, this cutoff point is 0.5. However, models trained on imbalanced data tend to result in all of the predicted values below 0.5, which is not useful.

Instead of using 0.5 for each model cutoff, the cutoff point will be chosen in a way that balances the number of correctly predicted funded and non-funded loans on the validation set. Section 4.1 states that maximizing the Improvement Metric is the main goal of the model. However, it is not practical to choose the cutoff point that only maximizes the Improvement Metric because this could result in a model that only classifies a small handful of leads as funded loans. In order to maximize the number of correctly classified leads, the cutoff point needs to balance the number of correctly classified funded loans as well as the number of correct negative classifications. These are both measured by the Sensitivity and Specificity, respectively. With this in mind, the following equation can be used to find the optimal cutoff point for each model on the validation set:

$$\text{Optimal Threshold} = \min\left((1 - \text{Sensitivity})^2 + (1 - \text{Specificity})^2\right) \tag{4.4}$$

This uses the Sensitivity and Specificity as previously determined in the section 4.1 and searches for the threshold value that minimizes the right side of equation 4.4. This method works to select the cutoff point that maximizes the number of funded loans correctly classified, while also minimizing the number of false-positive classifications. Once the optimal cutoff point is determined on the validation set, that same cutoff point will be used for model evaluation on the test set as well.

4.4 SMOTE Dataset

A secondary training dataset will be created using the SMOTE methodology for each feature set. It is crucial that SMOTE is not applied to either the validation set or the test set because this would then use synthetic entries for model evaluation, which leads to inaccurate results. SMOTE will only be applied to the training dataset to use for training each model variation on an artificially balanced dataset. This SMOTE training set will be created using the training set as designated in table 4.1, but will have a different distribution of entries. Rather than performing SMOTE once on the training set and using only the selected features from it, SMOTE will be performed once for each feature set, resulting in five different artificially balanced training sets. This is done so that the SMOTE algorithm doesn't select the nearest neighbor based on features that aren't important to the model.

Each SMOTE dataset was created using the SMOTE() function from the DMwR package in R [3]. When creating the SMOTE data, it is important to balance the ratio of undersampling and oversampling. Sampling too many artificial values or not sampling enough from the majority class can result in an augmented dataset that contains misinformation about which leads actually ended up funding. Finding the proper balance that didn't misconstrue the lead data required several attempts. The best ratio turned out to be using about 6% of the majority class and increasing the minority class by 575%. Even though there appears to be a large disparity between these ratios, the resulting augmented data contains lead funding trends comparable to that of the original dataset; except the augmented dataset is now balanced between the two classes.

The results of SMOTE using these sampling ratios on feature set 5 can be found in Appendix B. The resulting SMOTE datasets on the smaller feature sets contain similar results to those shown there. While the original training set has 247,013 entries, this new SMOTE dataset only has 18,998 entries. Most importantly, the distribution of the response variable in the SMOTE dataset is much more balanced. While the original training set showed only 0.56% of the total leads as funded loans, this new dataset shows 42.9% of the leads as funded loans.

This artificial balancing introduces some slight bias into the synthetic dataset, but the fund rates of the variables within each feature remain directionally consistent with the original data. Table B.1 shows the distribution of variables and fund rates within the SMOTE dataset compared to the original, unaltered dataset.

Even though the actual fund rates are drastically higher in the SMOTE dataset, all of the variables maintain the same trends in fund rate found in the original data. This indicates that the SMOTE dataset maintains much of the same information as the original data and should be valid for training.

CHAPTER 5

Model Results

5.1 Training Results

Each model is trained and evaluated twice – once on the original training dataset and again on the SMOTE training dataset. Each of these trained models is evaluated on the same validation set to compare their results. This is then repeated for each of the five feature sets, previously designated in table 3.24. The best performing model and feature set will be tested against the remaining test set to determine the final model results. As seen in table 4.1, the true fund rate of the validation set is 0.559%. This number will be used to calculate the Improvement Metric of each model on the validation set.

The following figures (figures 5.1 through 5.5) show the results of each model on each of the five feature sets for both the original data and the SMOTE datasets. Note that higher values are better for all three of these metrics. The table of exact values for all of these metrics can be found in Table C.1, located in Appendix C.

Feature Set 5 appears to have the highest values for the Improvement Metric, specifically for the Relogit model without SMOTE with a value of 0.6350. The Logit Model on Feature Set 5 has an Improvement Metric only 0.0038 below that of the Relogit model, which appears to be a similar pattern throughout all of the feature sets. This pattern is expected since Relogit is a modified form of Logistic Regression. The Relogit Model on Feature Set 5 has one of the lowest Specificity out of all the models tested (0.4448), but the highest Improvement Metric and Sensitivity.

The CatBoost model struggles to compete with the logistic regression models, as it is outperformed in almost every category. Results from the CatBoost model are not particularly

impressive on the original dataset, but show interesting results on the SMOTE dataset. CatBoost on the SMOTE dataset has the highest Specificity out of all the models tested at 0.7378. This is an impressive number, although it comes with a very high false-positive rate (Sensitivity of 0.3694) and the lowest Improvement Metric of 0.1690.

The models trained with the SMOTE dataset all show significant improvements in Specificity with a drop in Sensitivity and the Improvement Metric. It seems that the SMOTE dataset improves a model's ability to pick out the leads that do end up funding, although this results in more leads being incorrectly classified as funded loans.

Figure 5.1: Model Results on Feature Set 1.

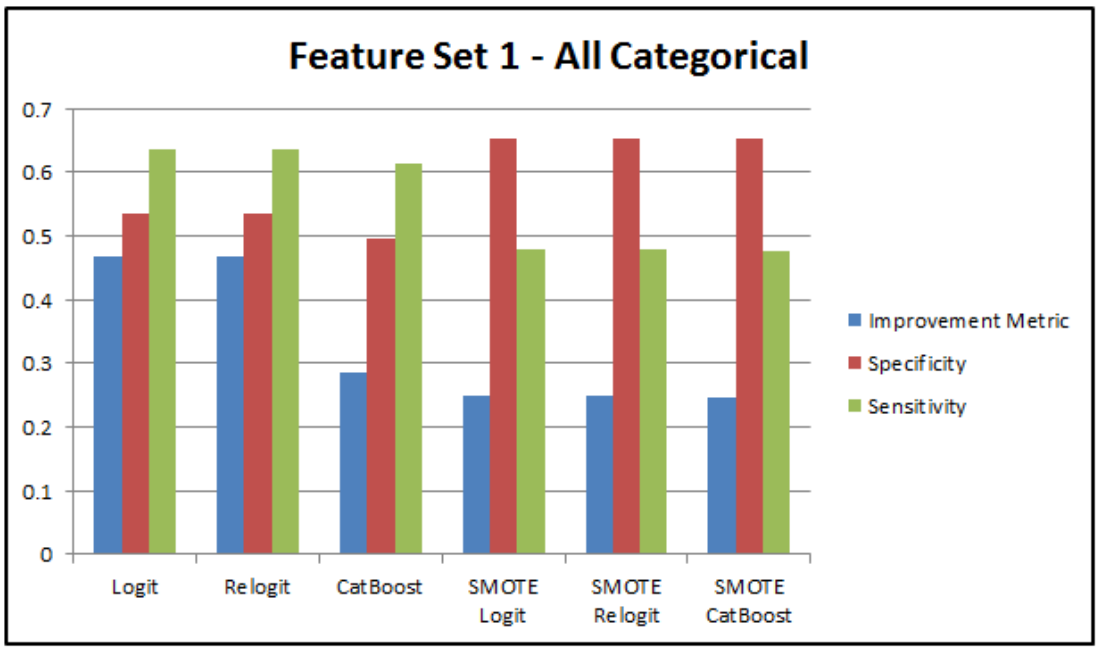


Figure 5.2: Model Results on Feature Set 2.

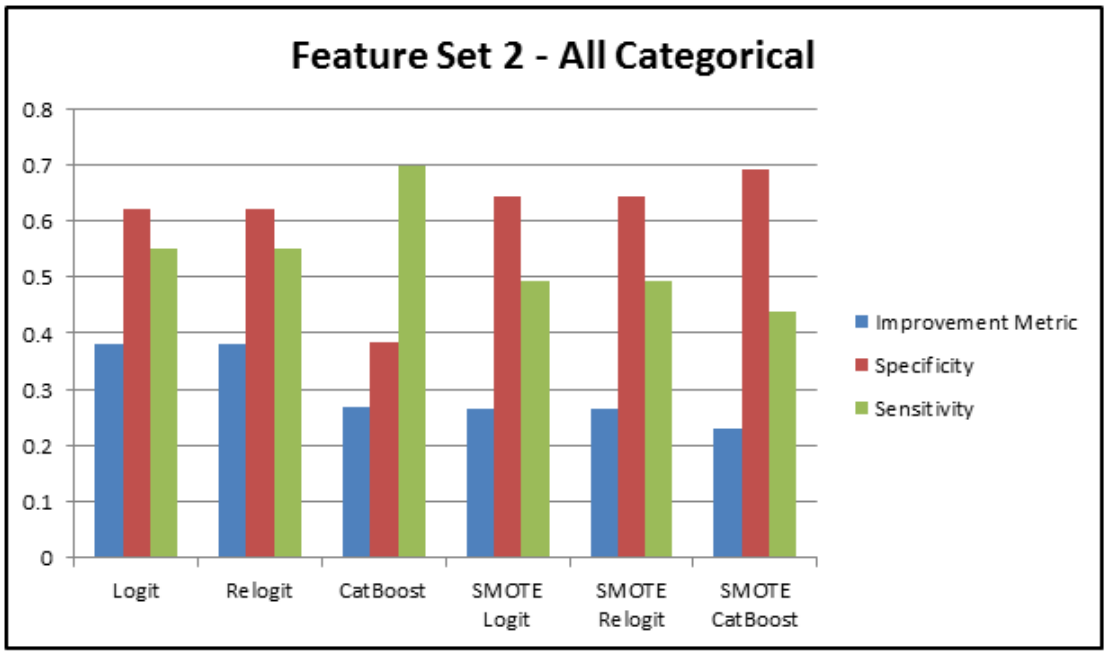


Figure 5.3: Model Results on Feature Set 3.

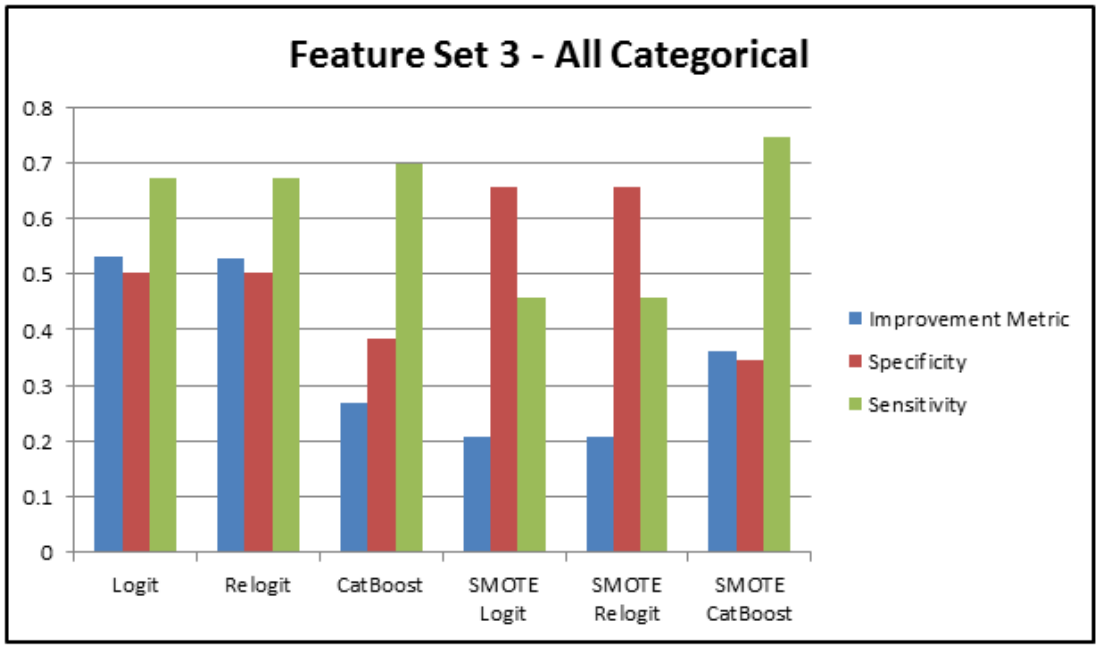


Figure 5.4: Model Results on Feature Set 4.

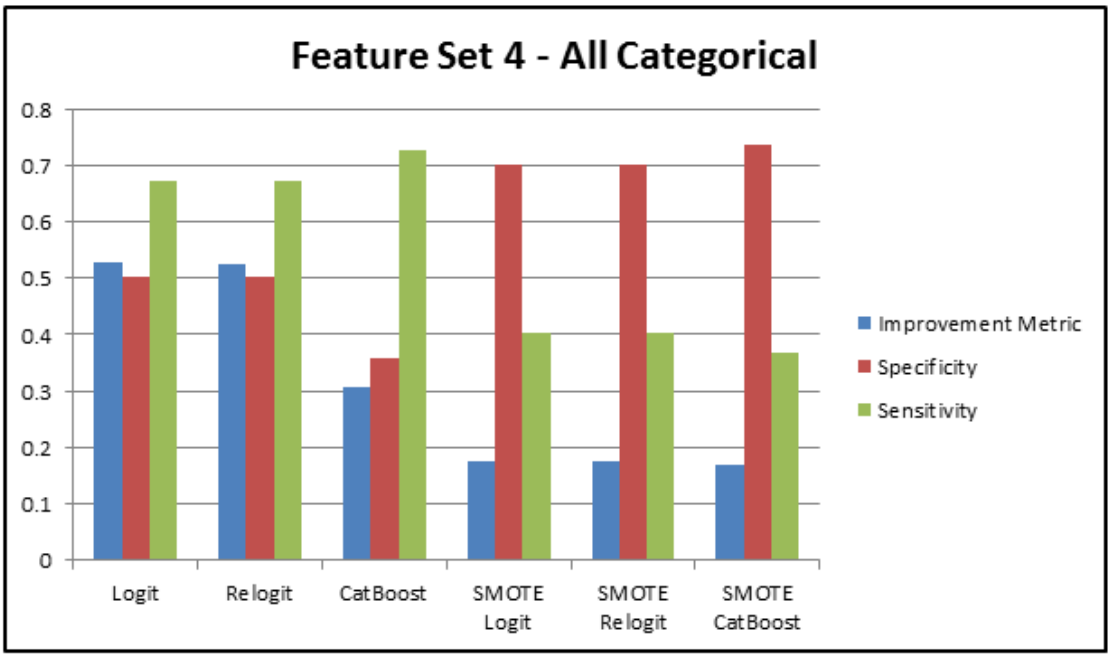
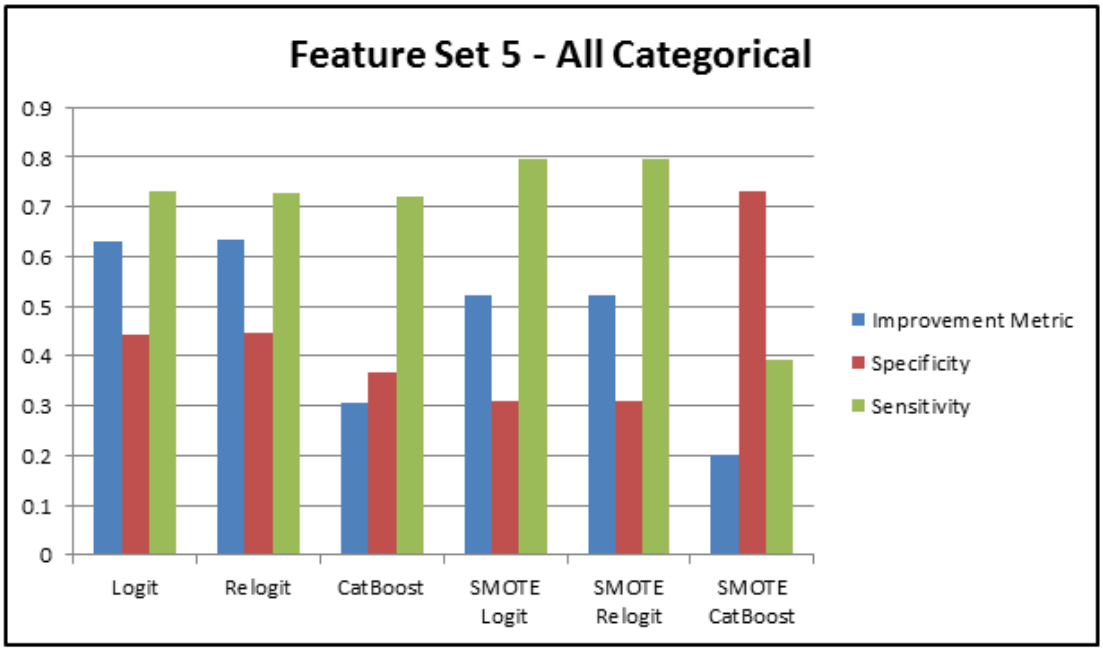


Figure 5.5: Model Results on Feature Set 5.



In addition to the feature sets shown above, the same test was run using the unaltered numeric features in the data. The feature sets were kept the same, with the only change being the numeric values for Loan Value, Add Cash, Mtg One Interest Rate, and Income were used instead of the grouped categorical values. The results of this test are shown in figures 5.6 through 5.10. The exact values for all of these metrics can be found in table C.2, located in Appendix C. Again, higher values are better for all three of these metrics.

None of the models trained on this data with numeric values outperformed the best model with all categorical variables. It should be noted that the CatBoost model trained on the original data with Feature Set 3 had an Improvement Metric of 0.6321, which is only 0.0029 below that best performing model on the data with all categorical values. This is also the best result for CatBoost out of all the models tested.

Similar trends are present in the results from this data compared to the results from the all-categorical data. Logit and Relogit models have very similar results, with Relogit having a slight edge, and Catboost tends to have higher Specificity and a lower Improvement Metric. SMOTE also performs similarly here, resulting in much higher Specificity at the cost of Sensitivity and the Improvement Metric.

Figure 5.6: Model Results on Feature Set 1.

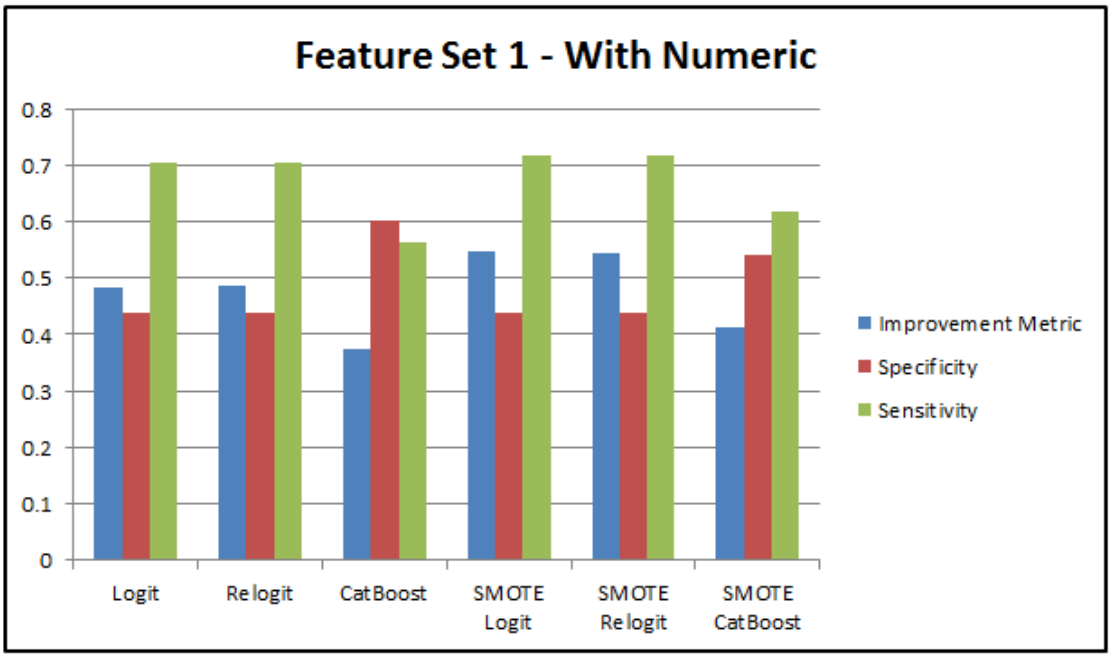


Figure 5.7: Model Results on Feature Set 2.

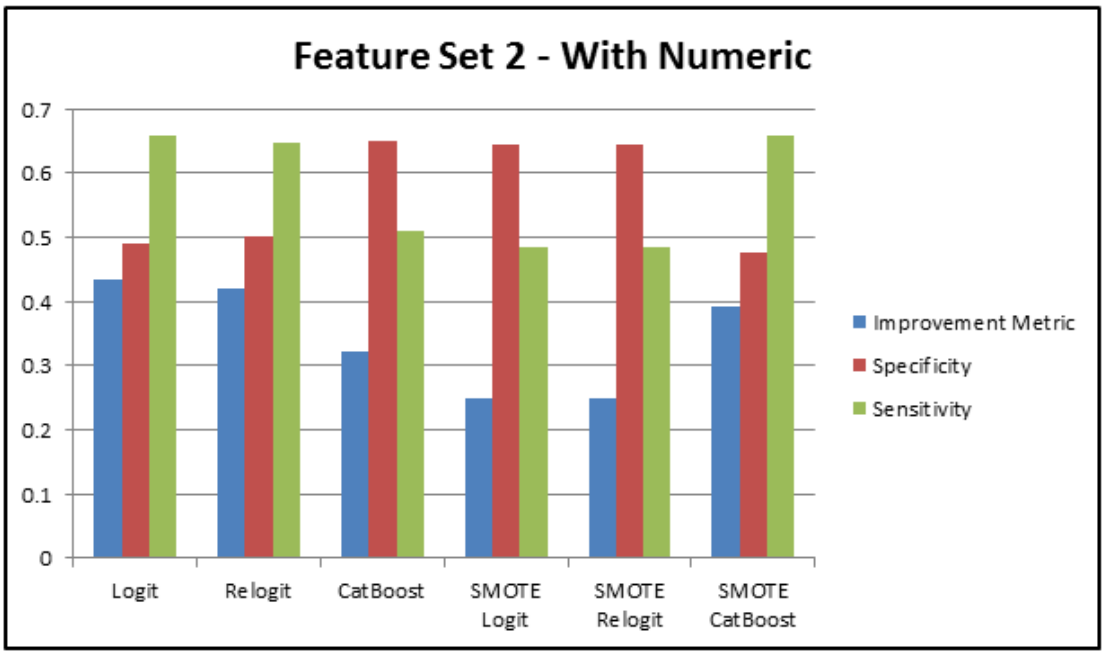


Figure 5.8: Model Results on Feature Set 3.

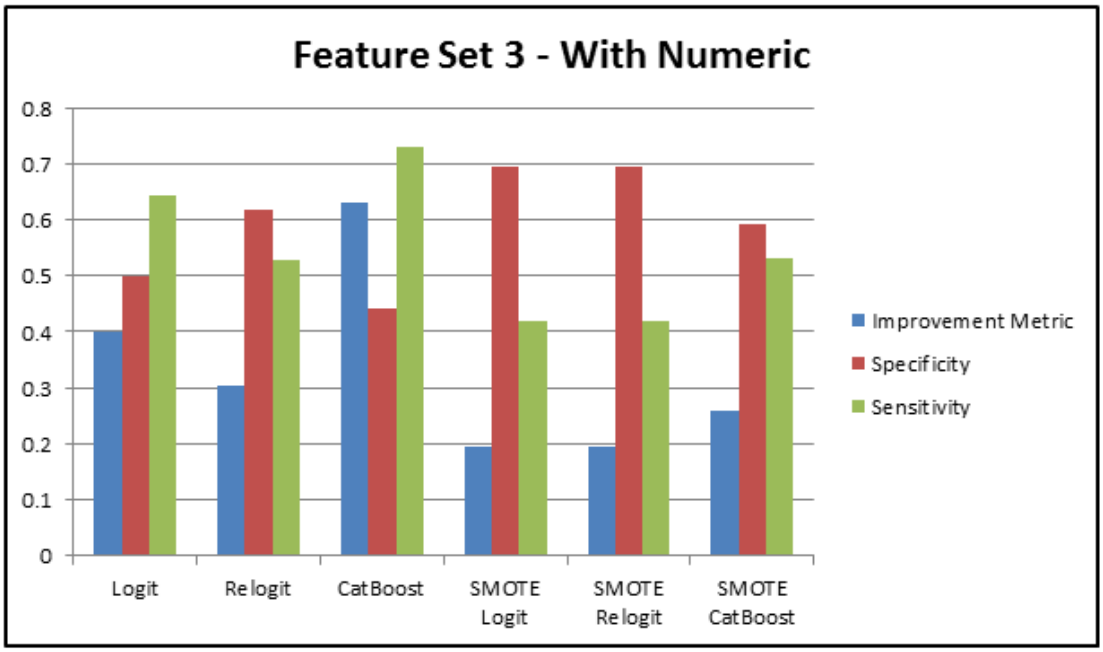


Figure 5.9: Model Results on Feature Set 4.

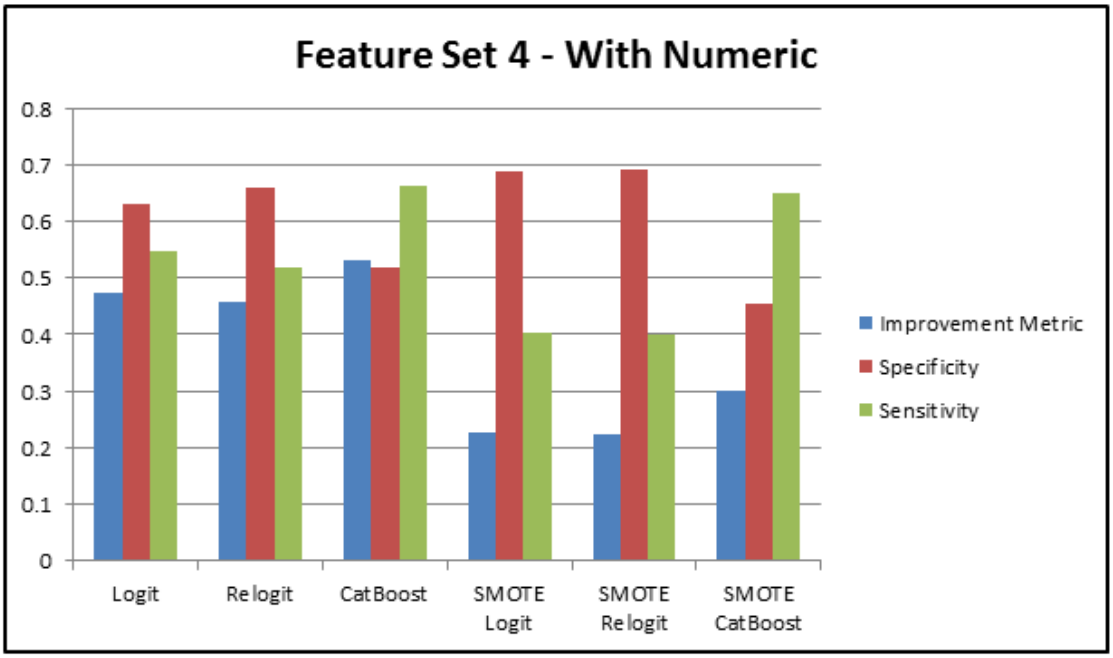
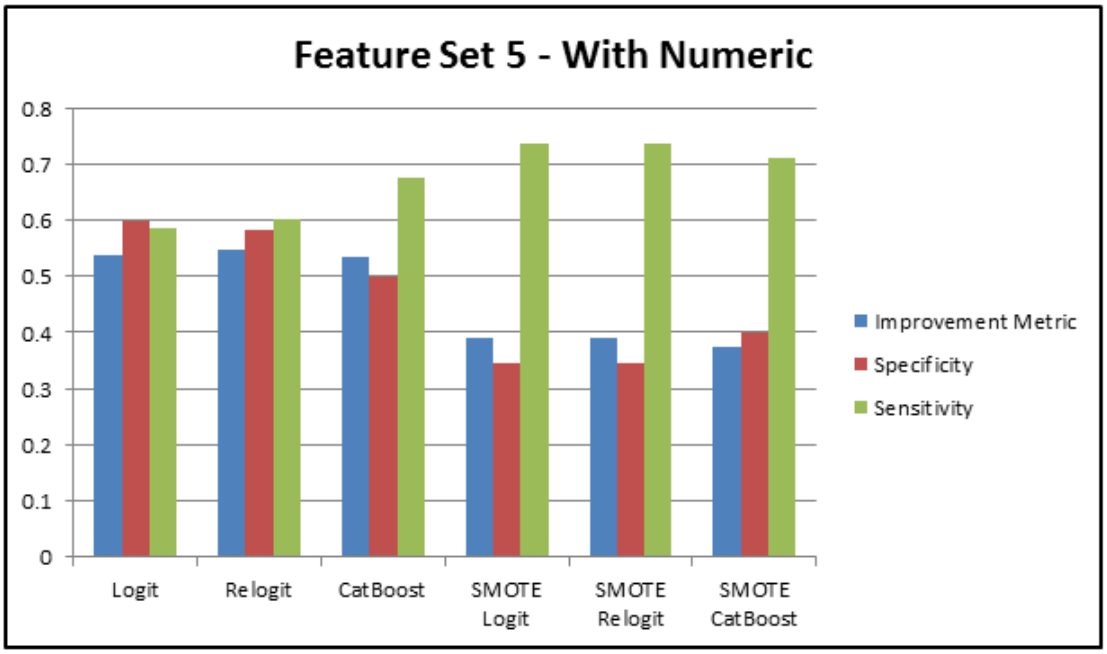


Figure 5.10: Model Results on Feature Set 5.



For the sake of this problem, the model with the highest Improvement Metric is the most desirable. Based off of these results, the best performing model is the Relogit model trained on the original all-categorical dataset with Feature Set 5. The Relogit model had an Improvement Metric of 0.6350, which slightly out-performed the Logit model at 0.6312. Relogit also had a slightly better Specificity than the Logit Model (0.4448 compared to 0.4433), although it had a slightly lower Sensitivity (0.7289 compared to 0.7292). While these differences might seem negligible, the effects on revenue are magnified due to the high volume of leads. So even though the Relogit model only marginally outperforms the Logit model, the Relogit model will have better results for the business over time.

5.2 Final Model Equation and Coefficients

The Relogit model trained on the original data had the highest Improvement Metric at 0.6350. This means that the model is able to identify a set of leads in the validation set that have a fund rate of 0.90%, compared to the original fund rate of 0.559%.

The resulting model equation from the Relogit Model is as follows:

$$P\left(\hat{y}=1\right)=\frac{1}{1+\exp\left(-\mathbf{x}\boldsymbol{\beta}\right)} \tag{5.1}$$

Where $P\left(\hat{y}=1\right)$ is the probability that a lead is predicted to be a funded loan, $\boldsymbol{\beta}$ represents bias adjusted coefficient estimates for each of the feature coefficients from feature set 5: $\tilde{\beta}_0$, $\tilde{\beta}_1$, ..., $\tilde{\beta}_{12}$, and $\mathbf{x}$ represents a series of input values from the lead: x_1, x_2, ..., x_{12} that correspond with each of the 12 features. Note that β_0 represents the intercept value, which has no corresponding input value from the lead.

Values for the coefficients in the final Relogit model are found in table 5.1. Each feature has all but one of its possible values listed in the table. The missing value is the baseline for that feature and has a coefficient estimate of 0. The model's prediction for a lead uses the corresponding coefficient estimates for the feature values present in that lead. So if a lead has a Credit Grade of Good, then the model uses the coefficient estimate for Credit Grade = Good, which is 0.111, and disregards the estimates for other values of Credit Grade.

These coefficients tend to directionally follow the trends discovered during the data exploration. The larger estimates tend to have stronger effects on the model's predictions. For example, the coefficient for Loan Value = \$650k+ is the lowest in the model at -0.885463. This indicates that leads with loan amounts higher than \$650k are unlikely to fund, even in the presence of other features that might make a loan more likely to fund. On the other hand, Add Cash = \$10k+ has the largest coefficient in the model at 0.682406, meaning leads with large add cash amounts are more likely to fund, even with other negative features present.

The two right-most columns in table 5.1 indicate the Z Value of the coefficient estimate and the calculated P-Value of that Z Value. This Z Value is calculated under the following null and alternative hypothesis:

$$h_0 = 0$$

$$h_1 \neq 0$$

Table 5.1: The coefficient estimates and standard errors for each feature in the final Relogit model.

Feature and Value:	**Coef. Est.**	**Std. Error**	**Z Value**	**Pr**$(>\|z\|)$
(Intercept)	-6.623	0.279	-23.687	<2e-16
Loan Value = $453-649k	-0.540	0.254	-2.214	0.034
Loan Value = $50-124k	0.576	0.060	9.525	<2e-16
Loan Value = $650k+	-0.845	0.382	-2.212	0.027
Add Cash = $10k+	0.658	0.065	10.054	<2e-16
Add Cash = $5-9k	0.127	0.075	1.688	0.091
Income = $70-99k	0.006	0.176	0.035	0.972
Income = Below $70k	0.241	0.166	1.453	0.146
FHA Eligible = True	0.143	0.064	2.233	0.026
Mortgage Lates = One	-0.470	0.144	-3.265	0.001
Mortgage Lates = Two+	-0.769	0.159	-4.812	1.49e-6
Credit Grade = Fair	-0.204	0.111	-1.83	0.067
Credit Grade = Good	0.111	0.059	1.878	0.060
Property Description = Single Family	0.538	0.186	2.891	0.004
Property Description = Townhome	0.376	0.252	1.495	0.135
Mtg One Interest Rate = 6-6.9%	0.203	0.119	1.695	0.090
Mtg One Interest Rate = 7%+	0.045	0.147	0.305	0.760
Loan Type = Fixed	0.266	0.133	2.00	0.046
VA Status = Yes	0.177	0.061	2.919	0.004
Purchase Period = Evening	-0.138	0.116	-1.185	0.236
Purchase Period = Morning	0.041	0.059	0.692	0.489
Purchase Period = Night	0.009	0.110	0.081	0.935
LTV Ratio = 76-85	0.001	0.071	0.020	0.984
LTV Ratio = 86-95	0.160	0.074	2.18	0.029
LTV Ratio = 96+	0.120	0.112	1.068	0.286

where β is the feature coefficient estimate. The Z value is then calculated as:

$$Z = \frac{\beta - 0}{\text{s.e.}} \tag{5.2}$$

where s.e. is the standard error of the coefficient estimate. This Z value is then used to calculate the P-Value as:

$$\text{P-Value} = 2 \times \Pr\left(0 \neq |z|\right) \tag{5.3}$$

This P-Value is used to determine the statistical significance of the coefficient. Those coefficients with a P-Value below 0.05 may be considered statistically significant, while those above 0.05 are not. This does not mean that the coefficient has no effect on the model's results, but rather is an indication of the influence of that coefficient within the model. The coefficients with lower P-Values have larger influence on the model's predictions. For example, Loan Value = \$50-124k has a P-Value of <2e-16, which clearly indicates that the coefficient is statistically significant. This feature has a coefficient of 0.576, which is one of the highest positive coefficients. The standard error on this coefficient is small at 0.06, indicating that this feature is consistently a strong predictor and has a good correlation with funded loans. The P-Value incorporates both the coefficient estimate and the standard error to evaluate whether that feature truly has an effect on the model or whether the coefficient could possibly have a value of 0 with no noticeable effect on the model. On the other hand, Purchase Period = Night has a P-Value of 0.935, indicating that this coefficient estimate is not statistically significant. This coefficient has an estimate of 0.009 with a standard error of 0.11, making it very possible that this feature has a coefficient estimate of 0, which would have no effect on the model's predictions.

Regarding these coefficient estimates, it is worth noting the following in the final Relogit model:

- **Loan Value**: The smaller loan amounts between \$50-124k have the highest estimate at 0.576, which confirms that lower loan amounts are more likely to fund. Consequently,

the higher loan amount categories of $453-649k and $650k+ both have high negative coefficients.

- **Add Cash**: Larger Add Cash values tend to fund more often, as demonstrated by the coefficient of 0.658 for Add Cash = $10k+. This is the largest positive coefficient, with a small standard error of 0.065, which indicates that this feature has the strongest influence out of all of the features in the model.

- **Income**: Leads from lower income areas tend to fund more often than higher income, although this relationship is not statistically significant for either $70-99k or Below $70k. So even though this feature improves the model's predictive power as a whole, it is not a consistent predictor of a lead's likelihood to become a funded loan.

- **FHA Eligible**: Leads that are FHA Eligible are a lower risk for lenders, which is clearly represented here as FHA Eligible has a statistically significant coefficient of 0.143.

- **Mortgage Lates**: Having any late mortgage payments tends to decrease the likelihood of a lead becoming a funded loan, as indicated by the negative coefficients. Interestingly, the Two+ category has one of the strongest negative coefficients in the model at -0.769 with a P-Value of 1.49e-06, indicating that this feature is a strong indicator that a lead will not become a funded loan.

- **Credit Grade**: The relatively few Fair credit leads are less likely to fund, with a coefficient of -0.204, while Good credit leads outperform Excellent credit with a coefficient of 0.111.

- **Property Description**: While both of these coefficients listed have a positive estimate, it is important to note that the baseline value is Multi-Family homes, with a coefficient estimate of 0, meaning they are much less likely to fund than other kinds of homes.

- **Mtg One Interest Rate**: It is interesting to note that the 6-6.9% interest rate category has a much larger coefficient estimate than the 7%+ category (0.203 compared

to 0.045). This would indicate that higher interest rates only improve the fund rate to a certain point (about 7%) at which they become less indicative of a lead funding.

- **Loan Type**: Fixed rate loans tend to fund more often than adjustable rate loans, although this is feature's P-Value of 0.046 is only barely under the cutoff of 0.05.

- **VA Status**: As expected, VA loans are more likely to fund, with a coefficient of 0.177 and a P-Value of 0.004.

- **Purchase Period**: The only value that really affects the model in any meaningful way is leads submitted in the Evening, which has a coefficient estimate of -0.138. It is interesting to note that Evening is the only value for Purchase Period that has a negative coefficient.

- **LTV Ratio**: The only statistically significant coefficient for LTV Ratio is that of 86-95, which has a P-Value of 0.029. It is interesting to note that 76-85 has a coefficient estimate of 0.001, which has practically no effect compared to most of the other coefficients in the model.

Even though some of the coefficient estimates are not statistically significant, they should not be removed from the final model because doing so decreases the performance metrics of the model.

5.3 Final Model Evaluation

With the Relogit model confirmed to be the best performing model on the validation set, the final evaluation is to see how it performs on the test dataset. The test dataset contains 25% of the full dataset and is used to evaluate the final results of the model that showed the best performance on the validation dataset. Table 5.2 shows the final results of the Relogit model predictions on the test dataset and table 5.3 shows the confusion matrix of the model's results.

Table 5.2: The final metric evaluation results for the Relogit Model on the test set.

Model	**Improvement Metric**	**Specificity**	**Sensitivity**
Relogit	0.6704	0.4623	0.7242

Table 5.3: The confusion matrix from the final results of the Relogit Model on the test set.

Relogit	**True Non-Fund**	**True Fund**
Predicted Non-Fund	88,958	357
Predicted Fund	33,878	307

The Relogit model has a higher Improvement Metric on the test set than on the validation set (0.6312 on the validation set compared to 0.6704 on the test set). The Specificity is also better on the test set, at 0.4623, which is good since it means the model can correctly predict almost half of the true funded loans. The final model is able to identify a set of leads that has a fund rate of 0.90%, which is a 67.04% improvement from the test set's original fund rate of 0.54%.

This may seem like a very minimal improvement over just randomly selecting leads from the test set, but in actuality, this is a very good model that provides a big opportunity for LeadPoint. There is such a myriad of factors that can influence whether a lead becomes a funded loan or not that even a 0.90% fund rate on a subset of leads is an impressive result. If it was easier to predict funded loans with much more accuracy, then lenders wouldn't waste their time and resources buying bulk sets of leads that only fund at 0.55%. The value of this model to LeadPoint can be quantified. Each lead has a set price of $10 in LeadPoint's marketplace. This Relogit model is able to pick out a subset that includes about 28% of all leads and this subset will have a fund rate that is 1.6704 times higher than any normal lead. LeadPoint has two apparent options of what to do with these leads.

First, they could raise the price on these leads, since they will be higher quality and likely

result in better returns for the lenders who purchase them. And based on the higher fund rate, they could charge up to 1.6704 times the regular price of a lead (up to $16.70/lead). By charging $16.70 for each of the leads predicted to fund, LeadPoint would immediately increase their revenue by about 19%. That is a huge gain from simply implementing this Relogit model into their lead generation process.

Second, they could use these higher quality leads to bring in a wider variety of lenders. The promise of high quality leads is enticing for all lenders, especially those who aren't equipped with an army of automatic phone dialers. The higher fund rate of the leads predicted to fund by the Relogit model would enable LeadPoint to sell to new lenders that they wouldn't be able to do business with otherwise. The additional revenue from this option is not as easy to quantify, but bringing on new lenders is a strategic initiative for LeadPoint and will result in additional revenue and company expansion over time.

CHAPTER 6

Modeling Outlook

6.1 Modeling Method Comparison

Even though the Relogit model showed the best performance on this data, it is far from perfect. First, it is only marginally better than the baseline Logit model, indicating that the Relogit methodology does not greatly improve the predictive power above what we would otherwise expect from logistic regression. In fact, the results from the Relogit model were identical to that of the Logit model for Feature Set 1 and Feature Set 2 when using the original data. When both models were trained on the SMOTE data, their results were identical for every feature set. This happened because the SMOTE process artificially balanced the data used for training. Without an imbalance in the data, the Relogit model's bias adjustment results in a value of 0, making the model identical to regular Logistic Regression.

It is unfortunate that the Relogit model's results do not differ more drastically from those of the Logit model, but it is good to see that the bias adjustment does ultimately give a slight edge to the Relogit model. This indicates that this model likely does improve predictions on rare event data over regular logistic regression. The improvement would probably be more visible on smaller sample sizes, as recommended by Zeng and King [5].

An example of how to implement Rare Event Logistic Regression can be found in Appendix D, including a sample dataset, example R code, and simple explanations of the code output.

The CatBoost model saw mixed results, although it was generally worse than the other two models. CatBoost worked well with SMOTE to result in higher Specificity values than any other model. It seems that the CatBoost model with SMOTE is best able to correctly

identify the leads that will eventually fund, but it does so at the cost of a lower Improvement Metric. The CatBoost model also seemed to have the best results on Feature Sets 3 and 4. This is in contrast to the Relogit and Logit models which both showed the best performance on Feature Set 5. This indicates that the Gradient Boosted Decision Trees built with CatBoost do not necessarily benefit from more features. CatBoost also performed consistently better with the numeric features included rather than the feature sets with all categorical variables.

SMOTE did not seem to have the desired effect on the models. In almost every case, the model trained on the SMOTE data had a lower Improvement Metric than the same model trained on the original data. On the other hand, the models trained on SMOTE tend to have higher Specificity than the models trained on the original data. This is likely due to the biases introduced in the SMOTE process. The new dataset is full of artificial leads, which may lead the models to put a heavier emphasis on some features. Then any leads that have those features would be predicted to fund, even though that may not always be accurate. This tends to result in a higher Specificity, even though there are more false-positives that lower the Improvement Metric.

This paper did not consider any neural networks or deep learning models, but these could provide a wide array of options that might help improve these results in future tests. Other methods of fitting Gradient Boosted Decision Trees were considered for this paper, including XGBoost and LightGBM, but CatBoost was chosen for its supposed shorter training times. Some preliminary tests on this data considered the use of Anomaly Detection, but these tests fell flat and did not produce significant results compared to those presented in this paper.

It is probable that this model could be improved upon in future iterations, but any model's performance will be hindered by the quality of the lead data. Further improving this model will likely depend more on increasing the quality of the training data than by implementing more intricate modeling methods.

6.1.1 Further Uses for Rare Event Logistic Regression

In this case, the Rare Event Logistic Regression was able to improve upon the results of regular logistic regression. This type of model might see improvements in other fields of study with rare events as well. For example, Zeng and King initially studied the occurrences of rare events in society; such as wars, disease, and political affairs [5]. They found that Rare Event Logistic Regression improved their ability to predict these types of events. According to Zeng and King, Rare Event Logistic Regression excels when the sample size is relatively small and when the positive class is found in less than 5% of the data. Based on their findings, it seems reasonable that Rare Event Logistic Regression could potentially improve predictions on any dataset that fits this description. Other areas that might benefit from experimenting with this type of modeling could include disease detection and treatment, natural disasters, terrorist activities, or even famine prediction.

6.2 Limitations

Data Limitations

All of the models tested are limited by the nature of the data. The most obvious limitation is that there are many more factors that influence whether a person refinances their mortgage that are not captured in this dataset. These factors could include debt-to-income ratio, geographic location, the age of the consumer's credit file, or even the current interest rate environment. Furthermore, there are a limited number of feature values in the current dataset. It is difficult to train an accurate model when thousands of leads have the same exact attributes but only a handful of them actually end up funding.

These limitations in the data could be partially mitigated by supplementing this dataset with third party data. For example, there are several companies that can provide a wealth of information about a consumer based on their home address. This data could include the consumer's age, their time at the residence, or other demographic information. Adding this supplemental data would help to build a profile of the type of consumer that is most likely

to refinance their loan.

Taking this idea one step further, if LeadPoint is able to ask for the consumer's social security number when they fill out their information, this could be used to do a soft-pull on the consumer's credit file. An official credit profile provides even more detailed information, including their actual credit score, their debt-to-income ratio, their other lines of credit, and even the history of their mortgage. All of this data would likely prove invaluable for use in predicting which leads are more likely to fund. LeadPoint is considering implementing both of these options in the future, which could pave the way for significant model improvements.

Study Limitations

This study is a great beginning for LeadPoint's goal to improve the quality of their mortgage leads. However, this project has several limitations that prevent better immediate results. First, this data is from a five month time period, which makes it difficult to account for any macroeconomic forces that might change throughout that time frame. For example, as national mortgage interest rates shift, consumers may gain or lose the benefits that refinancing brings. Future studies should attempt to shorten the time period from which the leads are selected, or include the weekly interest rates as a part of the model.

Second, it is important to consider that the data in these leads is all self-reported by the consumer. This means that it is likely that some of the data is incorrect, which may lead to incorrect results in the model. If a consumer indicates that they have Excellent credit, but actually have Poor credit, the lender has no choice but to reject their application. It is unclear how many consumers input wrong information, either intentionally or accidentally, but this is guaranteed to have an impact on which leads fund. Future studies will have to be creative in order to address this issue. One solution, also addressed in the data limitations, would be to pull the consumer's credit report, which guarantees the accuracy of their personal information. However, this is not an easy solution to implement by any means.

Finally, this study only considers refinance leads. LeadPoint also sells home purchase leads, which have drastically different fund rates and different features for use in a predictive

model. These home purchase leads not only have a lower fund rate than the refinance leads, but also require the consumer to both find a house to purchase and also qualify to receive a mortgage. Future studies should consider fitting a separate model for home purchase leads, which would enable LeadPoint to further increase their revenue and customer retention on both of their major products.

6.3 Multiple Classification Alternative

The original problem posed in this paper was to create a model that gives a binary classification of whether a lead will fund or not. Given the imbalanced nature of the data and the features, a binary classification is almost guaranteed to have sizeable error, as demonstrated here. However, by creating multiple classifications for the results instead of a binary "fund" or "not fund" prediction, the model becomes more versatile.

Consider the following: Figure 6.1 shows a histogram of the predicted values from the final Relogit model. The vertical red line marks the cutoff point used to classify leads above the cutoff as a "fund" and leads below the cutoff as "not fund". On the other hand, figure 6.2 shows the same histogram of values with two different cutoff points (one at 0.005 and the other at 0.1). In this instance, leads with prediction scores to the right of the blue line would be considered "High Tier" leads, leads in-between the blue and red lines would be considered "Mid Tier" leads, and leads below the red line would be considered "Low Tier" leads.

Rather than a prediction of whether a lead will fund or not, each lead would be placed into one of these categories as a ranking of the lead's quality. Then each subset of leads will have a different fund rate and the leads in each category can be priced accordingly. The fund rates of each subset can be seen in table 6.1, along with the Improvement Metric as calculated from the test dataset's fund rate of 0.54%. This method classifies the leads according to their propensity to become a funded loan and each classification can be priced according to the Improvement Metric: High-Tier leads are more expensive to account for the 1.1% fund rate and Low-Tier leads are cheaper to offset the 0.38% fund rate. The result is

a more robust methodology for pricing the leads according to quality, although it does not specifically answer the original question of whether a specific lead will fund or not.

Figure 6.1: Histogram of the prediction values from the Relogit Model on the test set with the original cutoff point.

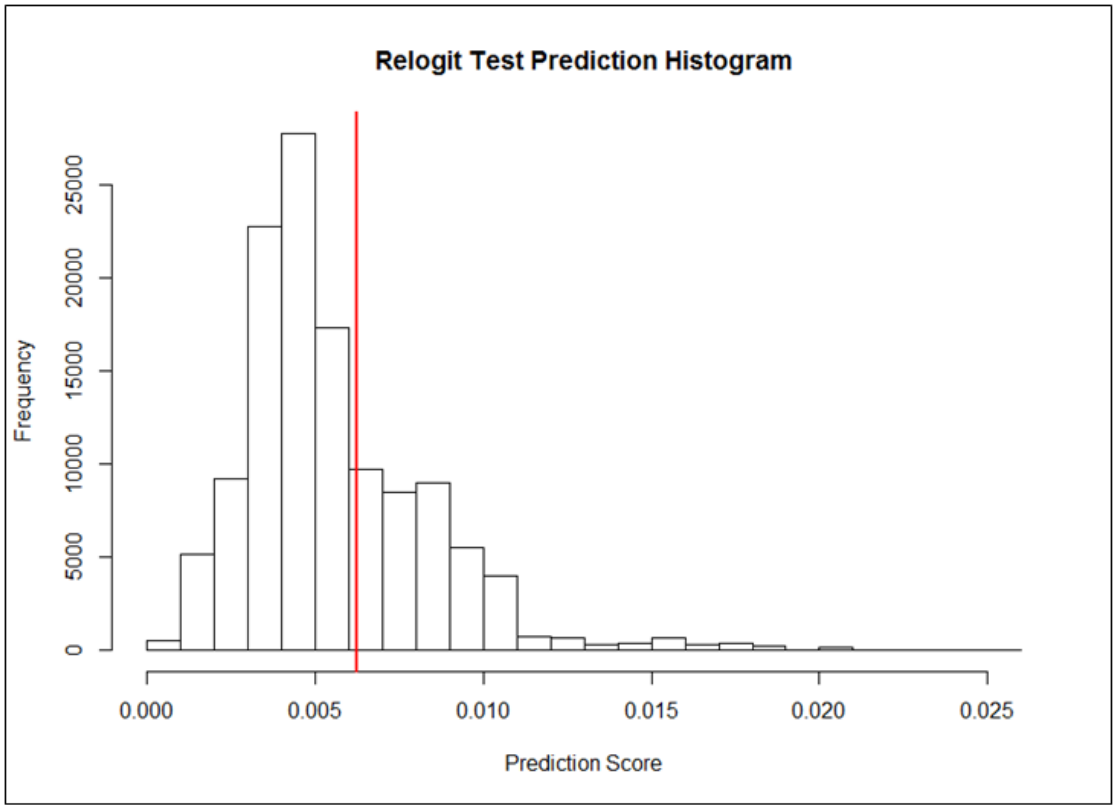
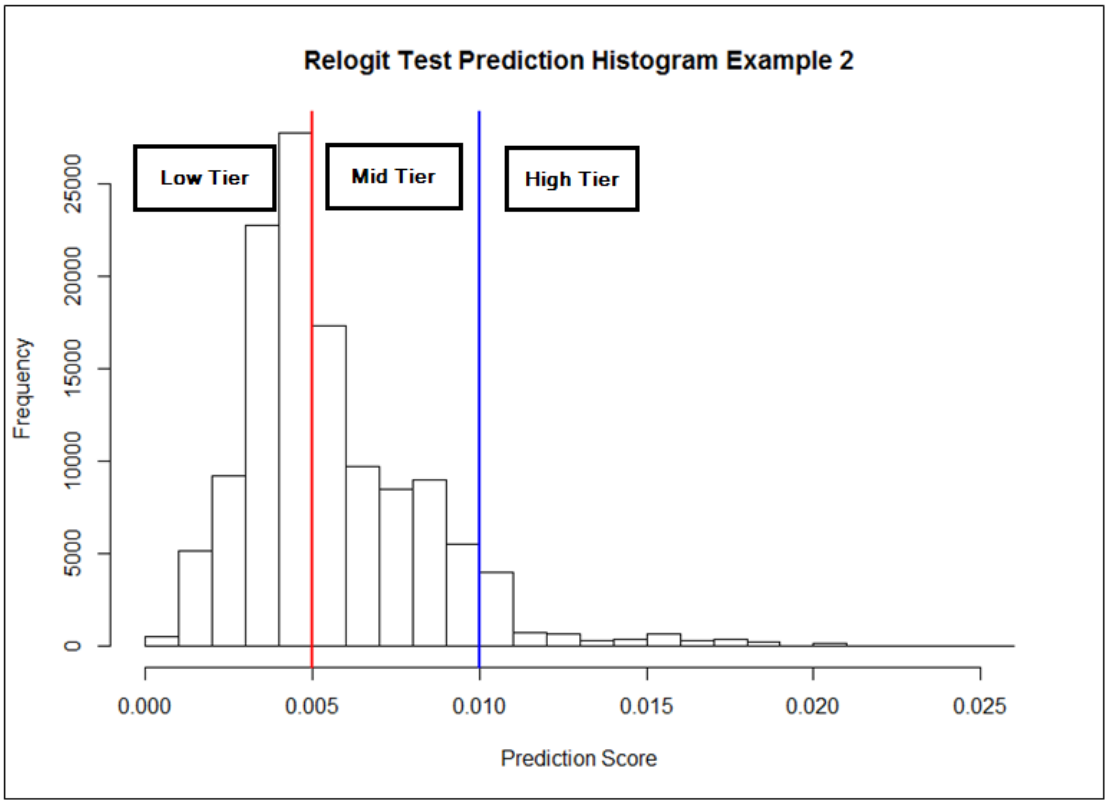


Table 6.1: Table showing the fund rate and Improvement Metric for each of the tiered classifications.

Category	**Total Leads**	**Fund Rate**	**Improvement Metric**
High-Tier	7,901	1.10%	1.048128
Mid-Tier	50,078	0.66%	0.229422
Low-Tier	65,483	0.38%	-0.3012424

Figure 6.2: Histogram of the Relogit Model prediction values with two cutoff points to classify leads based on their predicted fund rate.



CHAPTER 7

Conclusion

Ultimately, the Rare Event Logistic Regression Model (Relogit) trained on the original dataset had the best results for predicting whether a given lead was likely to fund or not. The Relogit model was chosen to be the best because it had the highest Improvement Metric on the validation set out of all the models tested. The Relogit model's predictions on the test set resulted in an Improvement Metric of 0.6704, a Specificity of 0.4623, and a Sensitivity of 0.7242. Where the test set has a fund rate of 0.54%, this model is able to identify a subset of leads (28% of the total) that have a fund rate of 0.90%. This is a very good result and this model will bring immediate tangible benefits to LeadPoint's business.

Since the model is able to identify leads that are 1.6704 times more likely to fund than usual, LeadPoint can sell them at a higher price or sell them to specifically chosen lenders to strategically expand their business. By selling the leads at the higher price, this model could potentially bring an immediate 19% increase in company revenue.

This model is constrained by limitations within the training dataset, but still manages to extract useful information out of the data to accurately identify just under half of the leads that will end up funding. Furthermore, the proposed ranking methodology would allow for more flexibility in lead pricing according to a lead's propensity to fund and seems to be the most robust application for this model. Future iterations of this model should consider any data supplementation that might improve the quality of the data.

Utilizing the rare event logistic regression model will improve the experience of the lenders purchasing these mortgage leads and will also bring in additional revenue to LeadPoint Inc. The ranking methodology could improve this even further by allowing each lender the opportunity to choose the efficiency and price of the leads that they purchase.

Appendix A

Logistic Regression

Logistic regression is a modified form of linear regression, except the result is interpreted as the probability of an event occurring. In traditional linear regression, the response variable is modeled as follows:

$$\hat{y} = \beta_0 + \beta 1 x_1 + \ldots + \beta_n x_n \tag{A.1}$$

which demonstrates the relationship between the response variable and the explanatory variables in a linear fashion. The result, ŷ, is a predicted value for the outcome that can take any value in the sample space of the response variable. However, logistic regression only has two values for the response variable: 0 or 1. In this case, a special function is used called the logistic function, which transforms any input into a value between 0 and 1. The logistic function is defined as follows:

$$f(x) = \frac{1}{1 + \exp(-x)} \tag{A.2}$$

This function works such that as x increases toward infinity, the result is closer to 1 and as x decreases toward negative infinity, the result is closer to 0. This function is useful because all probabilities must take on a value between 0 and 1, just like the output of the logistic function. Applying the logistic function to the linear regression model results in the following:

$$P(\hat{y} = 1) = \frac{1}{1 + \exp(-x\beta)} \tag{A.3}$$

Where β represents a series of input weights, β_0, β_1, ..., β_x. The Logistic function forces the result of the equation to be a value between 0 and 1, which represents the probability that the event in question occurs. With this prediction, a cutoff point is used to determine whether $\hat{y}$ is considered as predicting the event to occur or not. This cutoff is usually designated as 0.5; so if $\hat{y}$ is above 0.5 then the event is predicted to occur and if $\hat{y}$ is below 0.5, then the event is not predicted to occur. This cutoff point can be changed depending on the nature of the data and the predictions.

The input weights in a logistic regression model (β_0, β_1, etc.) cannot be interpreted in the same way as weights in a linear regression model. Because of the logistic function, the weights no longer affect the response variable linearly but the decision boundary and all level sets remain linear. Picking up from equation A.3, the following equation sheds some light on how to properly interpret these weights:

$$\begin{aligned}
P(\hat{y}=1) &= \frac{1}{1+\exp(-(\boldsymbol{b}))} \\
1+\exp(-\boldsymbol{b}) &= \frac{1}{P(\hat{y}=1)} \\
\exp(-\boldsymbol{b}) &= \frac{1}{P(\hat{y}=1)} - 1 \\
\exp(-\boldsymbol{b}) &= \frac{1}{P(\hat{y}=1)} - \frac{P(\hat{y}=1)}{P(\hat{y}=1)} \\
\exp(-\boldsymbol{b}) &= \frac{1-P(\hat{y}=1)}{P(\hat{y}=1)} \\
\exp(-\boldsymbol{b}) &= \frac{P(\hat{y}=1)}{1-P(\hat{y}=1)} \\
\boldsymbol{b} &= \left(\frac{P(\hat{y}=1)}{1-P(\hat{y}=1)}\right) \\
\boldsymbol{b} &= \left(\frac{P(\hat{y}=1)}{P(\hat{y}=0)}\right)
\end{aligned} \tag{A.4}$$

Where $\boldsymbol{b}$ is the weighted sum of input features: $\beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n$. It should also be noted that in the considered model, the ratio of $\frac{P(\hat{y}=1)}{P(\hat{y}=0)}$ is strictly positive, thus the ratio is well-defined.

The probability of an event occurring divided by the probability of the same event not

occurring is called the odds ratio. The last line in equation A.4 shows the model weights equal to the log of the odds ratio of the event occurring, also known as the log odds. Thus, the logistic regression model is a linear model for the log odds of the event occurring. This means that a change in x_i by one unit increases the log odds ratio by the value of the corresponding weight, β_i.

Logistic regression is solved by finding a set of parameters, $\boldsymbol{b}$, that maximizes the log likelihood of the data, expressed as follows:

$$\mathcal{L} = \sum_{i=1,\, y_i=1}^{N} \log P(x^{(i)}) + \sum_{i=1,\, y_i=0}^{N} \log \left(1 - P(x^{(i)})\right) \tag{A.5}$$

Where N is the number of individual observations, (X, y) is a set of observations, X is is a K + 1 by N matrix of inputs with each column representing an observation (and the first row is 1), K is the number of parameters, y is an N-dimensional vector of responses, and each (x_i, y_i) represents an individual observation.

Solving for the weights is done using gradient optimization (such as gradient descent or other second order methods like Newton's method). This is a root-finding algorithm that iteratively produces better approximations for the roots of a real-valued function. Starting with a vector valued function $y = f(b)$, Newton's method attempts to optimize b so that $f(b_{optimal}) = 0$. It starts with an initial value, b_0, and takes the Taylor expansion of f around b_0 as:

$$f(b_0 + \Delta) \approx f(b_0) + f'(b_0)\Delta \tag{A.6}$$

Note that f' is a matrix, which is the Jacobian of first derivatives of f with respect to b. Set the left side of the equation to 0 and solve for Δ as follows:

$$\Delta_0 = -[f'(b_0)]^{-1} f(b_0) \tag{A.7}$$

Where $[f'(b_0)]^{-1}$ is the matrix inverse of $f'(b_0)$. This requires that the inverse matrix exists.

After solving for Δ_0, this value is used to update the estimate for b such that:

$$b_1 = b_0 + \Delta_0 \tag{A.8}$$

And this process is continued until convergence of $\boldsymbol{b}$.

In the case of logistic regression, f is the gradient of the log-likelihood and the Jacobian is the Hessian, or the matrix of second derivatives of the log-likelihood function with respect to $\boldsymbol{b}$. First, we calculate the gradient of the log-likelihood with respect to $\boldsymbol{b}$:

$$\begin{aligned}
\nabla_b \mathcal{L} &= \sum_{i=1,\, y_i=1}^{N} \frac{P_i'}{P_i} x_i - \sum_{i=1,\, y_i=0}^{N} \frac{P_i'}{1-P_i} x_i \\
&= \sum_{i=1,\, y_i=1}^{N} \frac{P_i\,(1-P_i)}{P_i} x_i - \sum_{i=1,\, y_i=0}^{N} \frac{P_i\,(1-P_i)}{1-P_i} x_i \\
&= \sum_{i=1,\, y_i=1}^{N} (1-P_i) x_i - \sum_{i=1,\, y_i=1}^{N} P_i x_i \\
&= \sum_{i=1}^{N} \left[y_i (1-P_i) - (1-y_i) P_i \right] x_i \\
&= \sum_{i=1}^{N} \left(y_i - P_i \right) x_i
\end{aligned} \tag{A.9}$$

Where P_i is shorthand for $P(x_i)$. Then, we can use this to determine the Hessian:

$$\begin{aligned}
H &= \frac{\partial}{\partial \boldsymbol{b}} \nabla_b \mathcal{L} \\
&= \frac{\partial}{\partial \boldsymbol{b}} \sum_{i=1}^{N} \left(y_i - P_i \right) x_i \\
&= -\sum_{i=1}^{N} x_i \nabla_b P_i \\
&= -\sum_{i=1}^{N} x_i P_i (1-P_i) x_i^T \\
&= \boldsymbol{X W X}^T
\end{aligned} \tag{A.10}$$

Where $\boldsymbol{W}$ is a diagonal matrix of the derivatives P_i', and the ith column of $\boldsymbol{X}$ corresponds to the ith observation. With this in mind, each iteration of solving for Δ is calculated as:

$$\Delta_k = \left(\boldsymbol{X}\boldsymbol{W}_{\boldsymbol{k}}\boldsymbol{X}^T\right)^{-1}\boldsymbol{X}\left(\boldsymbol{y} - \boldsymbol{P}_k\right) \tag{A.11}$$

Where $\boldsymbol{y}$ represents the vector of observed responses, $\boldsymbol{W}$ is the current matrix of the derivatives P_i', $\mathbf{P}_k$ is the vector of probabilities calculated by the current estimate of $\boldsymbol{b}$, and k is the current iteration. At the end of each iteration, the value of Δ_k is the solution to a weighted least squares problem such that the result is the difference between a given observed response and its current estimated probability of being true. This Δ_k value is then used to update the parameter values from equation A.8 at the k-th step of the optimization loop.

Appendix B

Feature Distribution of the SMOTE Dataset

Table B.1: Table showing the difference in the distribution and fund rate between the Original and SMOTE datasets for each variable in Feature Set 5.

	Original			SMOTE		
Values	**Leads**	**Funded**	**Fund Rate**	**Leads**	**Funded**	**Fund Rate**
	Purchase Period					
Morning	123,718	707	0.57%	9,343	4,176	44.7%
Afternoon	85,684	473	0.55%	6,616	2,822	42.7%
Evening	18,838	89	0.47%	1,573	542	34.5%
Night	18,773	102	0.54%	1,466	602	41.1%
	Loan Value					
$50-124k	58,259	478	0.82%	5,583	2,966	53.1%
$125-453k	178,078	870	0.49%	12,749	4,971	39.0%
$453-649k	6,633	16	0.24%	386	124	32.1%
$650k+	4,043	7	0.17%	280	81	28.9%
	Credit Grade					
Excellent	95,713	497	0.52%	7,431	3,174	42.7%
Good	127,678	767	0.60%	9,866	4,314	43.7%
Fair	23,603	107	0.45%	1,701	654	38.4%
	LTV Ratio					
70-75	249,321	1,380	0.55%	9,233	3,920	42.4%
76-85	115,068	586	0.51%	4,545	1,837	40.4%

86-95	90,394	527	0.58%	3,635	1,664	45.8%
96+	39,242	221	0.56%	1,585	721	45.5%
	Add Cash					
$0-5k	150,727	696	0.46%	10,445	3,768	36.1%
$5-9k	46,579	261	0.56%	3,329	1,315	39.5%
$10k+	49,707	414	0.83%	5,224	3,059	58.6%
	Property Description					
Multi-Family	9,818	30	0.31%	2,465	631	25.6%
Single-Family	229,342	1,307	0.57%	14,687	6,807	46.3%
Townhome	7,853	34	0.43%	1,846	704	38.1%
	Mtg. One Interest Rate					
2-5.9%	227,719	1,245	0.55%	16,036	6,016	37.5%
6-6.9%	11,065	76	0.69%	1,441	1106	76.8%
7%+	8,229	50	0.61%	1,521	1020	67.1%
	Mortgage Lates					
None	437,994	2,505	0.57%	13,212	6,557	49.6%
One	25,816	112	0.43%	2,465	841	34.1%
Two+	30,215	97	0.32%	3,321	744	22.4%
	VA Status					
No	183,566	975	0.53%	13,214	5,111	38.7%
Yes	63,447	396	0.62%	5,784	3,031	52.4%
	FHA Eligible					
False	171,752	873	0.51%	12,088	4,573	37.8%
True	75,176	497	0.66%	6,910	3,569	51.6%
	Loan Type					
Adjustable	14,961	61	0.41%	3,896	1,229	31.5%
Fixed	232,052	1,310	0.56%	15,102	6,913	45.8%
	Income					

$100k+	10,780	39	0.36%	666	216	32.4%
$70-99k	48,997	204	0.42%	3,954	1,435	36.3%
Below $70k	187,236	1,128	0.60%	14,378	6,491	45.1%

Appendix C

Model Training Results Tables

First, the results of each model trained on the feature sets with all categorical features.

Table C.1: The results of each model trained on the original dataset followed by the SMOTE dataset with all categorical variables.

Feature Set 1	**Improvement Metric**	**Specificity**	**Sensitivity**
Logit	0.4670	0.5361	0.6355
Relogit	0.4670	0.5361	0.6355
CatBoost	0.2849	0.4963	0.6143
SMOTE Logit	0.2500	0.6539	0.4776
SMOTE Relogit	0.2500	0.6539	0.4776
SMOTE CatBoost	0.2461	0.6539	0.4759
Feature Set 2	**Improvement Metric**	**Specificity**	**Sensitivity**
Logit	0.3793	0.6200	0.5514
Relogit	0.3793	0.6200	0.5514
CatBoost	0.2697	0.3829	0.6989
SMOTE Logit	0.2652	0.6421	0.4932
SMOTE Relogit	0.2652	0.6421	0.4932
SMOTE CatBoost	0.2291	0.6922	0.4376
Feature Set 3	**Improvement Metric**	**Specificity**	**Sensitivity**
Logit	0.5300	0.5037	0.6717
Relogit	0.5288	0.5037	0.6715
CatBoost	0.2697	0.3829	0.6989

SMOTE Logit	0.2076	0.6554	0.4579
SMOTE Relogit	0.2076	0.6554	0.4579
SMOTE CatBoost	0.3623	0.3461	0.7465

Feature Set 4	**Improvement Metric**	**Specificity**	**Sensitivity**
Logit	0.5270	0.5037	0.6717
Relogit	0.5248	0.5037	0.6706
CatBoost	0.3079	0.3594	0.7257
SMOTE Logit	0.1758	0.7025	0.4031
SMOTE Relogit	0.1758	0.7025	0.4031
SMOTE CatBoost	0.1690	0.7378	0.3694

Feature Set 5	**Improvement Metric**	**Specificity**	**Sensitivity**
Logit	0.6312	0.4433	0.7292
Relogit	0.6350	0.4448	0.7289
CatBoost	0.3067	0.3652	0.7210
SMOTE Logit	0.5222	0.3093	0.7974
SMOTE Relogit	0.5222	0.3093	0.7974
SMOTE CatBoost	0.2023	0.7305	0.3931

And the following table shows the results of each model trained on the feature sets with the numeric features included.

Table C.2: The results of each model trained on the original dataset followed by the SMOTE dataset using the four numeric variables instead of their categorical groupings.

Feature Set 1	**Improvement Metric**	**Specificity**	**Sensitivity**
Logit	0.4831	0.4389	0.7049
Relogit	0.4864	0.4389	0.7055
CatBoost	0.3734	0.6009	0.5634
SMOTE Logit	0.5460	0.4389	0.7170
SMOTE Relogit	0.5453	0.4389	0.7169
SMOTE CatBoost	0.4116	0.5420	0.6169
Feature Set 2	**Improvement Metric**	**Specificity**	**Sensitivity**
Logit	0.4351	0.4904	0.6591
Relogit	0.4196	0.5022	0.6470
CatBoost	0.3220	0.6495	0.5096
SMOTE Logit	0.2496	0.6451	0.4845
SMOTE Relogit	0.2496	0.6451	0.4845
SMOTE CatBoost	0.3907	0.4757	0.6587
Feature Set 3	**Improvement Metric**	**Specificity**	**Sensitivity**
Logit	0.4007	0.5007	0.6433
Relogit	0.3024	0.6171	0.5269
CatBoost	0.6321	0.4404	0.7311
SMOTE Logit	0.1938	0.6951	0.4183
SMOTE Relogit	0.1936	0.6951	0.4182
SMOTE CatBoost	0.2603	0.5920	0.5309
Feature Set 4	**Improvement Metric**	**Specificity**	**Sensitivity**
Logit	0.4749	0.6319	0.5461

Relogit	0.4561	0.6597	0.5200
CatBoost	0.5322	0.5169	0.6636
SMOTE Logit	0.2255	0.6898	0.4029
SMOTE Relogit	0.2246	0.6921	0.4005
SMOTE CatBoost	0.3001	0.4551	0.6506

Feature Set 5	**Improvement Metric**	**Specificity**	**Sensitivity**
Logit	0.5373	0.5995	0.5870
Relogit	0.5477	0.5810	0.6025
CatBoost	0.5337	0.4978	0.6764
SMOTE Logit	0.3912	0.3449	0.7373
SMOTE Relogit	0.3911	0.3449	0.7372
SMOTE CatBoost	0.3742	0.3991	0.7102

Appendix D

Sample Data and Example R Code of Rare Event Logistic Regression

This section is meant to give an example of Rare Event Logistic Regression; including a sample of the data used in this paper and some example R code for implementing this method. For a brief review, Rare Event Logistic Regression is a form of logistic regression developed by Zeng and King [5] that implements a method of bias reduction to improve model accuracy. This method is specifically meant to be used on data where the positive class occurs in less than 5% of the sample, also referred to as rare event data. Rare Event Logistic Regression was created to improve predictions on when these rare events are most likely to occur. In this paper, it was used to predict funded loans that occur in only 0.55% of the sample data. This method might also be considered for use in predicting the presence of rare diseases, the occurrence of natural disasters, the likelihood of armed conflicts escalating, or any other instance where the event in consideration seldom occurs.

The data depicted in figure D.3 is an example of the lead data used in this paper. It includes all 14 data points, including the funding status of the lead.

Implementing Rare Event Logistic Regression is made simple through the use of the 'Zelig' package in R [6]. This package contains a function that fits a regression model using Zeng and King's bias adjustment. This function is called zelig() and it is capable of fitting several different types of models, one of which is Rare Event Logistic Regression. The code for importing the package, the data, and fitting a relogit model is given below:

```
# Begin by importing the required packages.
# Zelig is used for fitting the model and caret is used to split the data.
```

```
library(Zelig)
library(caret)

# Import the data into R.
leads <- read.csv('lead_data.csv', header=TRUE, stringsAsFactors = FALSE)

# Use the following to preview the data, if desired.
head(leads)

# It is important to split the data into two parts - one training set and one
    test set.
set.seed(15)
index <- createDataPartition(leads$Funding, p = 0.75, list = FALSE)
train <- leads[train_index,]
test <- leads[-train_index,]

# The zelig() function works like the glm() function in base R.
# By specifying model = 'relogit', it will fit a Rare Event Logistic Regression
    model to the dataset.
# The first argument, "Funding ~ ." specifies that "Funding" is the response
    variable and that all of the other variables should be used in fitting the
    model.
relogit_model <- zelig(Funding ~ . , data = train, model = 'relogit')
```

Once the model is fit, there are tools in R that can be used to explore how well the model performs. Specifically, it is important to consider the following:

- What are the feature coefficients?
- What is the model's Improvement Metric?
- What is the model's Sensitivity?

- What is the model's Specificity?

The code below indicates how to answer all of those questions about the model.

```
# This function displays a summary of the feature coefficients:
summary(relogit_model)

# To calculate the model performance metrics, it is best to create a confusion
    matrix of the model results.
predictions <- predict(relogit_model, newdata = test, type = 'response')[[1]]
labels <- ifelse(predictions >= 0.5, 1, 0)
confusion_matrix <- table(labels, test$Funding)
print(confusion_matrix)

# The three model metrics are calculated as follows:
improvement_metric <- ((confusion_matrix[4] / (confusion_matrix[2] +
    confusion_matrix[4]) ) / mean(test$Funding) - 1)
sensitivity <- confusion_matrix[4] / (confusion_matrix[3] + confusion_matrix[4])
specificity <- confusion_matrix[1] / (confusion_matrix[2] + confusion_matrix[1])
improvement_metric
sensitivity
specificity
```

An example of the output from the summary() function is shown in figure D.1. In this figure, the area of focus is on the information below "Coefficients:". This will list each of the feature coefficients, as well as information about the actual coefficient estimate and the standard error of that estimate. This is the same data that was explored in section 5.2, which gives more information about how to interpret each of these numbers. At a high level, features with larger estimates tend to cause the model to predict that the lead will fund, features with negative estimates cause the model to predict that the lead will not fund, and features with estimates of 0 have little-to-no effect on the model's prediction.

An example of the resulting confusion matrix can be seen in figure D.2, with labels to indicate what each of the sections represent. The values in this table are then used to calculate all of the model metrics. The metrics calculated here are also described in section 4.1, but they will be repeated here for convenience.

Improvement Metric - This is a measure of the model's precision compared to the true fund rate of the test data set.

Specificity - This is a measure of how many true funded loans that the model correctly predicted to fund.

Sensitivity - This is a measure of how many true non-funding leads that the model accurately predicted.

The code above gives all of the information necessary to fit a basic Rare Event Logistic Regression model and measure how well it performed. This code can easily be modified to test out different feature sets and can also be modified to fit models to entirely different sets of data.

Figure D.1: An example of how the model summary looks with the feature coefficient estimates.

```
> summary(relogit_m1)
Model:

Call:
z5$zelig(formula = Convert ~ ., data = dude)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.2420  -0.1183  -0.0987  -0.0867   3.6743

Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)             -6.6344048  0.2804756 -23.654  < 2e-16
purch_periodevening     -0.1382447  0.1162782  -1.189 0.234474
purch_periodmorning      0.0415002  0.0597277   0.695 0.487166
purch_periodnight        0.0085904  0.1096625   0.078 0.937562
BID_LOAN_VALUE453-649k  -0.5434770  0.2544146  -2.136 0.032664
BID_LOAN_VALUE50-124     0.5768906  0.0604942   9.536  < 2e-16
BID_LOAN_VALUE650k+     -0.8569189  0.3829637  -2.238 0.025247
CRED_GRADEFAIR          -0.2013937  0.1114886  -1.806 0.070855
CRED_GRADEGOOD           0.1121308  0.0589670   1.902 0.057225
```

Figure D.2: An example of what the output histogram might look like, with each column and row labeled.

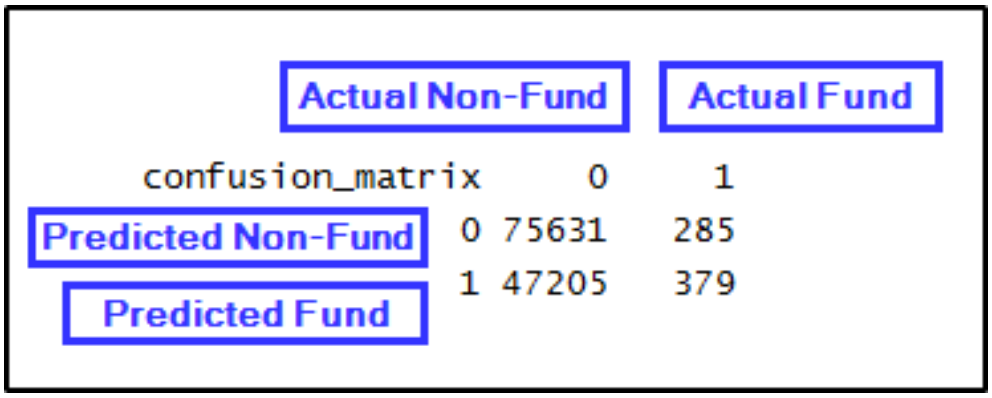


Figure D.3: A snippet of the lead data from LeadPoint.

Loan Value	Credit Grade	LTV Ratio	Add Cash	Property Description	Mtg One Interest Rate	Second Mortgage	Mortgage Lates	VA Status	FHA Eligible	Loan Type	Income	Funding
50-124	GOOD	70-75	$10k+	Single_Fam	2-5.9	NO	NONE	YES	FALSE	FIXED	Below$70k	0
125-453k	GOOD	76-85	$0-$5k	Single_Fam	2-5.9	NO	NONE	NO	FALSE	FIXED	Below$70k	0
125-453k	GOOD	76-85	$5-$9k	Single_Fam	2-5.9	NO	NONE	NO	FALSE	FIXED	Below$70k	0
125-453k	EXCELLENT	76-85	$0-$5k	Single_Fam	2-5.9	YES	NONE	YES	FALSE	FIXED	Below$70k	1
125-453k	GOOD	76-85	$0-$5k	Single_Fam	6-6.9	NO	NONE	NO	FALSE	FIXED	Below$70k	0
125-453k	EXCELLENT	86-95	$0-$5k	Single_Fam	2-5.9	NO	NONE	YES	FALSE	FIXED	Below$70k	0
125-453k	GOOD	76-85	$0-$5k	Single_Fam	2-5.9	NO	Two+	NO	FALSE	FIXED	Below$70k	0
125-453k	EXCELLENT	96+	$0-$5k	Single_Fam	2-5.9	NO	NONE	YES	FALSE	FIXED	Below$70k	0
125-453k	EXCELLENT	70-75	$0-$5k	Single_Fam	2-5.9	NO	NONE	NO	TRUE	ADJUSTABLE	Below$70k	0
125-453k	GOOD	86-95	$5-$9k	Single_Fam	2-5.9	NO	NONE	NO	FALSE	FIXED	$70-99k	0
125-453k	EXCELLENT	70-75	$0-$5k	Single_Fam	2-5.9	NO	NONE	NO	TRUE	FIXED	Below$70k	0
125-453k	GOOD	76-85	$0-$5k	Single_Fam	2-5.9	NO	NONE	NO	FALSE	FIXED	Below$70k	0
125-453k	GOOD	86-95	$5-$9k	Single_Fam	2-5.9	NO	NONE	YES	FALSE	FIXED	Below$70k	1
125-453k	EXCELLENT	70-75	$0-$5k	Single_Fam	2-5.9	NO	NONE	NO	TRUE	FIXED	Below$70k	0
50-124	GOOD	70-75	$0-$5k	Single_Fam	2-5.9	NO	NONE	NO	FALSE	FIXED	Below$70k	0

REFERENCES

[1] The Board of Governors of the Federal System. Mortgage debt outstanding. https://www.federalreserve.gov/data/mortoutstand/current.htm.

[2] Kevin W. Bowyer, Nitesh V. Chawla, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *CoRR*, abs/1106.1813, 2011.

[3] L. Torgo. *Data Mining with R, learning with case studies*. Chapman and Hall/CRC, 2010.

[4] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2018.

[5] Langche Zeng, Gary King, and Micahael Tomz. Relogit: Rare events logistic regression. *Journal of Statistical Software*, 08, 02 2003.

[6] Christine Choirat, James Honaker, Kosuke Imai, Gary King, and Olivia Lau. *Zelig: Everyone's Statistical Software*, 2018. Version 5.1.6.1.

[7] Yandex. Catboost. https://tech.yandex.com/catboost/.

[8] Anna Veronika Dorogush, Andrey Gulin, Gleb Gusev, Nikita Kazeev, Liudmila Ostroumova Prokhorenkova, and Aleksandr Vorobev. Fighting biases with dynamic boosting. *CoRR*, abs/1706.09516, 2017.

[9] Anna Veronika Dorogush, Vasily Ershov, and Andrey Gulin. Catboost: gradient boosting with categorical features support, 2018.

[10] Federal Communications Commission. Fcc actions on robocalls, telemarketing. https://www.fcc.gov/general/telemarketing-and-robocalls.

[11] SecureRights. Home equity quiz. https://www.homeequityquiz.com/?formFlowConfigId=1811&estprg=1&viewType=FULL#2.